

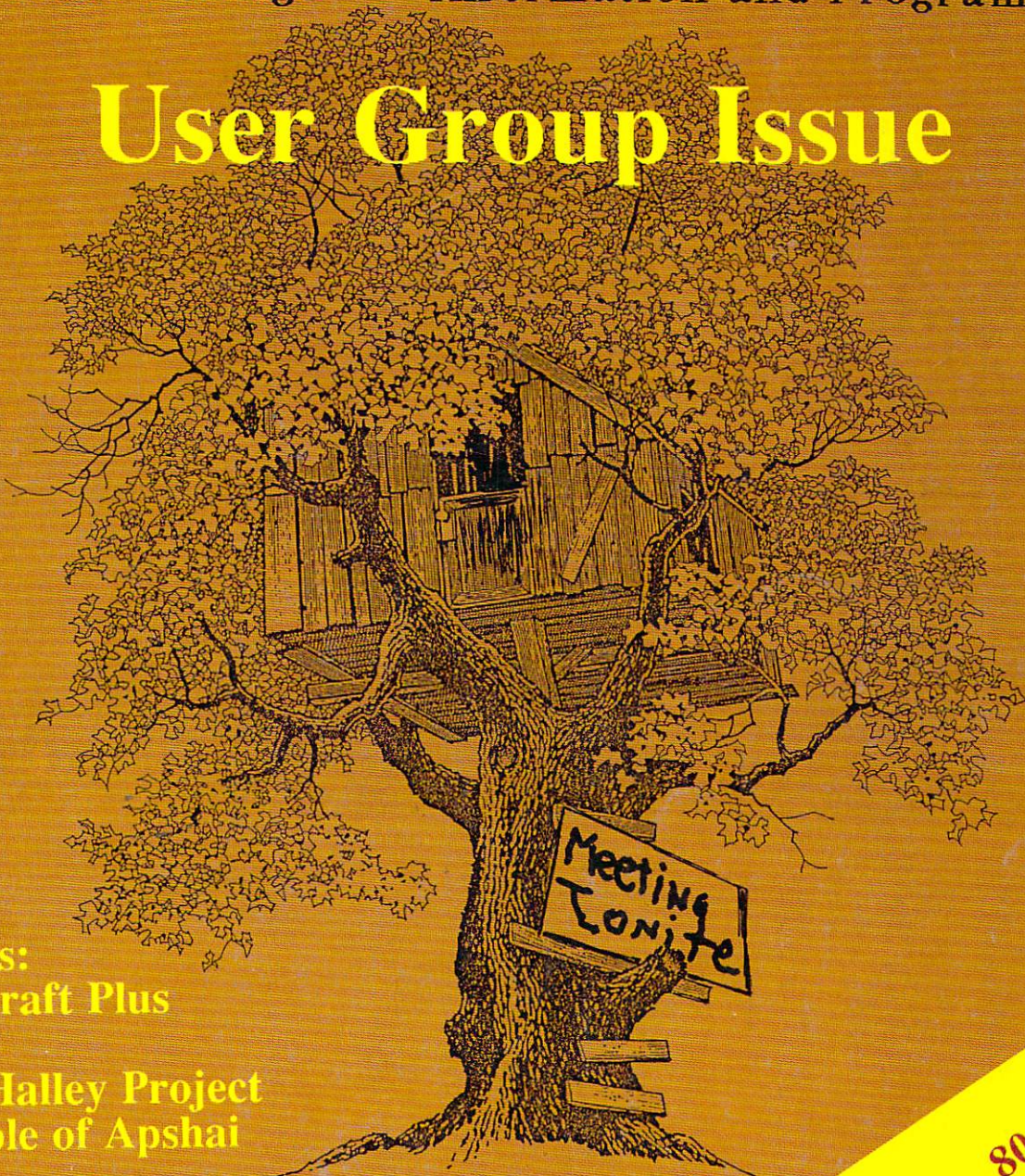
A Catalog of AMICUS & Fred Fish  
Public Domain Software

# Amazing Computing™

Printed July 1986  
Vol.1/ Number 6  
U.S.A. \$2.50  
Canada \$3.50

Commodore Amiga™ Information and Programs

## User Group Issue



Reviews:  
Textcraft Plus  
Flow  
The Halley Project  
Temple of Apshai

80 Pages!!!



|                |                                                                                                                            |                                                               |                                                                                                                                     |                                                       |                                                               |                            |                                      |       |                                                                                                                                                                                          |                                                                                 |                                                                                                                                                               |                                                                                                                   |                                                                                                                       |                                                     |                                                                       |                        |                                                                                                                 |                                               |                                                                                                                                                                                                                                                                                      |                                                                                                                                                         |                                                                                                                                                                                |                                                                                     |                                                   |                        |                                                                                                                        |                                                                     |                                                                                                                                                                                       |                                                                                                    |                                                                                                                                |                                                                        |                                             |         |                                             |             |                                                                    |                       |                                                                  |                |                                                                      |                                 |                                                                        |                 |                                                                                                                                                            |                                                               |                                                                                                                 |                                                        |                                                                              |                                       |                                                                                                                                    |                                                                                    |                                                                                                                                                                                                                                                                                                                |                                                                                                                                          |                                                                                                     |                                  |                                                                                                                                                                                                     |                                                                                           |                                                                                                                                                                                                   |                                                                                       |                                                                                     |                                   |                                                              |                             |                                                                                                                 |                                      |                                                                                                                                                                                                                                                                                                                                                                 |                                                                                                                                                                                                      |                               |                                                                                                 |                                                   |                                                                                                                         |                                                    |                                                                                                                                                |                                                                        |                |  |
|----------------|----------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------|---------------------------------------------------------------|----------------------------|--------------------------------------|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|-----------------------------------------------------------------------|------------------------|-----------------------------------------------------------------------------------------------------------------|-----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------|------------------------|------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------|---------------------------------------------|---------|---------------------------------------------|-------------|--------------------------------------------------------------------|-----------------------|------------------------------------------------------------------|----------------|----------------------------------------------------------------------|---------------------------------|------------------------------------------------------------------------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|--------------------------------------------------------|------------------------------------------------------------------------------|---------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-----------------------------------|--------------------------------------------------------------|-----------------------------|-----------------------------------------------------------------------------------------------------------------|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------|-------------------------------------------------------------------------------------------------|---------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------|----------------|--|
| <b>Alabama</b> | Alex Data Systems<br>Command Computers<br>Madison Books & Computers<br>Piedmont Computer Shop<br>Universal Computer System | Huntsville<br>Birmingham<br>Madison<br>Fayetteville<br>Mobile | <b>Georgia</b><br>Future System<br>Michael Donnellan<br>Software South Inc.<br>The 64 Store<br><b>Hawaii</b><br>Computer House Inc. | Augusta<br>Decatur<br>Savannah<br>Atlanta<br>Honolulu | <b>Iowa</b><br>Century Systems<br>Micro Computer Applications | Des Moines<br>Marshalltown | <b>Idaho</b><br>ASL Computer & Video | Nampa | <b>Illinois</b><br>BA Business Computers<br>Computer Revolution<br>ComputerLand O'Hare<br>Computers Land of Niles<br>Data Microcomputers Inc.<br>The Memory Expansion<br>Unique Computer | Batavia<br>Champaign<br>Park Ridge<br>Niles<br>Naperville<br>Normal<br>Sterling | <b>Indiana</b><br>Burkat Computer Ctr.<br>Bytrex Computer<br>Computer Corner<br>Computer People Inc.<br>Ging Chaney<br>Micro Computers Inc.<br>Vons Computers | South Bend<br>Elkhart<br>Fort Wayne<br>Mishawaka<br>Michigan City<br>Greenfield<br>Indianapolis<br>West Lafayette | <b>Kansas</b><br>Data Bank Corp.<br>J & L Electronics<br>MidKanas Computers<br>Midwest Computers<br>Thorold Computers | Wichita<br>Liberal<br>Newton<br>Manhattan<br>Topeka | <b>Kentucky</b><br>MicroAge Computer Store<br>Resully World Computers | Lexington<br>Lexington | <b>Louisiana</b><br>Modern Business Machines<br>Software Center Interm.<br>Software Mart<br>The Computer Clinic | Metairie<br>Metairie<br>Metairie<br>Lafayette | <b>Massachusetts</b><br>Club Computer<br>Computerland<br>E.J. Wurster Co.<br>General Computer Store<br>HCS Computer Store<br>LCA Video & Computer Ctr.<br>Memory Location<br>Omni Tek Computers<br>Pioneer Valley Data Equip.<br>Tech Computer Store<br>The Bit Bucket<br>Tycom Inc. | Cambridge<br>Leominster<br>Boston<br>Framingham<br>Marshfield<br>Norwood<br>Wellesley<br>Tewksbury<br>Amherst<br>Cambridge<br>West Newton<br>Pittsfield | <b>Maryland</b><br>Computer Crafters<br>ComputerWorld<br>Compusision Vision Ctr.<br>Micro Computer Center<br>Software Advantage<br>The Logical Choice Inc.<br>Waldorf Computer | Wheaton<br>Ocean City<br>Columbia<br>Baltimore<br>Rockville<br>Baltimore<br>Waldorf | <b>Maine</b><br>Computer Barn<br>Valley Computers | Damariscotta<br>Auburn | <b>Michigan</b><br>Basic Computer Center<br>Bits Bytes-Nibbles<br>Chelsea Computer<br>Computer Supply Co.<br>CompuSoft | Grand Rapids<br>Petoskey<br>Ann Arbor<br>Traverse City<br>Ridgeland | <b>Minnesota</b><br>Computer Outlets<br>Computer Place<br>Computer Specialties Plus<br>"Computers, etc."<br>JMI Software<br>MCD of Hibbing Inc.<br>Online Computers<br>Specialists IM | Roseville<br>Minneapolis<br>Monticello<br>Eagan<br>Maple Grove<br>Hibbing<br>Minnetonka<br>Hopkins | <b>Missouri</b><br>Associated Computer Ser.<br>Brandt Computers<br>Computer Concepts<br>Interstate Replay Ltd.<br>Systems Plus | Springfield<br>Kansas City<br>Columbia<br>Chesterfield<br>Independence | <b>Mississippi</b><br>Enterprises Unlimited | Jackson | <b>Montana</b><br>Appligen Computer Systems | Great Falls | <b>North Carolina</b><br>Digit<br>TDS Computers<br>Triad Computers | Camboro<br>Greensboro | <b>North Dakota</b><br>Computer Associates<br>The Computer Store | Fargo<br>Minot | <b>Nebraska</b><br>BULL'S EYE<br>Double E Electronics<br>My Com Inc. | Lincoln<br>Omaha<br>Plattsmouth | <b>New Hampshire</b><br>Computer Mart of N.H.<br>Diversified Computers | Nashua<br>Keene | <b>New Jersey</b><br>Family Computer Centres<br>Family Computer Centres<br>Family Computer Centres<br>Family Computing Centres<br>Morris County Stationers | Chester<br>Ridgewood<br>South Orange<br>Ridgewood<br>Cherwood | <b>New Mexico</b><br>Academy Computers<br>New Horizon Computer Sys.<br>Page One Newsstand<br>Technical Concepts | Albuquerque<br>Alamogordo<br>Albuquerque<br>Las Cruces | <b>Nevada</b><br>Century 23 Inc.<br>Computer World<br>ComputerLand of Carson | Las Vegas<br>Las Vegas<br>Carson City | <b>New York</b><br>Byte Shop<br>Castle Computer<br>CIA Software Center<br>Computer Centre<br>Computer Outlet<br>Lenoir's Computers | New York<br>Merrick<br>Latham<br>Flushing<br>White Plains<br>Jamestown<br>New York | <b>Ohio</b><br>Computer Network<br>Kalamazoo<br>Earthrise Micro Systems<br>Lakes Consumer Electronics<br>Micro Center<br>Microwave Magic<br>North Coast Programming<br>Quality Computer Applications<br>Telego<br>Saxton Computer Center<br>Software Centre Intern<br>Software and More<br>Software Connection | Columbus<br>Dayton<br>Akron<br>Columbus<br>Fairfield<br>Willoughby<br>Toledo<br>Sandusky<br>North Olmsted<br>Cincinnati<br>Mayfield Hts. | <b>Oklahoma</b><br>Colonial Video & Computer<br>P C Tech<br>Second Hand Software<br>Video-Comp Inc. | Tulsa<br>Oklahoma City<br>Lawton | <b>Oregon</b><br>Clackamas Computers<br>Clackamas Computers<br>Computer Business Sys.<br>Computer Business Sys.<br>IB Computers<br>Software Center<br>Software Express<br>Univ. of Oregon Bookstore | Beaverton<br>Clackamas<br>Portland<br>Tigard<br>Portland<br>Beaverton<br>Eugene<br>Eugene | <b>Pennsylvania</b><br>Alpha Omega EBE<br>Basic Computer Systems<br>Data Softique Computware<br>East Coast Software<br>Pittsburgh Computer Store<br>The Electronic Boutique<br>Triangle Computers | Warren<br>Hermitage<br>Pittsburg<br>Pittsburgh<br>Pittsburgh<br>Norristown<br>Indiana | <b>Rhode Island</b><br>International Computer Serv.<br>Micro Limits<br>Micro Limits | Johnston<br>Smithfield<br>Warwick | <b>South Carolina</b><br>Avcom International Inc.<br>Horizon | West Columbia<br>Greenville | <b>South Dakota</b><br>Computer Business Center<br><b>Tennessee</b><br>ComputerLab of Memphis<br>Software First | Rainier City<br>Memphis<br>Nashville | <b>Texas</b><br>Byte by Byte<br>C.E.M. Corporation<br>Colonial Video<br>Computer Age<br>Computer Age<br>Computer Magic<br>Computer Revelations<br>Computer Time Inc.<br>Metropolitan Computer<br>Micro Search<br>University Educational Sys.<br>Softex<br>Software Centre<br>Software Library<br>Texas Terminal<br>The Computer Experience<br>The Computer Stop | Austin<br>Brazoria<br>Houston<br>Houston<br>Dallas<br>Austin<br>Houston<br>Marshall<br>Richardson<br>Houston<br>Dallas<br>Watauga<br>Dallas<br>Wichita Falls<br>Fort Worth<br>San Antonio<br>Abilene | <b>Utah</b><br>Computers Plus | <b>Virginia</b><br>Colony Ltd.<br>Discovery<br>Family Computer Center<br>Virginia Micro Systems | Richmond<br>Falls Church<br>Fairfax<br>Woodbridge | <b>Washington</b><br>APPLE CO-OP<br>Com Soft Computers<br>Computers + Inc.<br>Micro Age Computer Store<br>Programs Plus | Renton<br>Everett<br>Tacoma<br>Bellevue<br>Seattle | <b>Wisconsin</b><br>Corlinton Computers<br>Computer Software Center<br>Morm's Computer<br>Software Shoppe<br>Southeastern<br>TMW Software Inc. | Racine<br>Milwaukee<br>Stevensport<br>Fond Du Lac<br>Madison<br>Wausau | <b>Wyoming</b> |  |
|----------------|----------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------|---------------------------------------------------------------|----------------------------|--------------------------------------|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|-----------------------------------------------------------------------|------------------------|-----------------------------------------------------------------------------------------------------------------|-----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------|------------------------|------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------|---------------------------------------------|---------|---------------------------------------------|-------------|--------------------------------------------------------------------|-----------------------|------------------------------------------------------------------|----------------|----------------------------------------------------------------------|---------------------------------|------------------------------------------------------------------------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|--------------------------------------------------------|------------------------------------------------------------------------------|---------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-----------------------------------|--------------------------------------------------------------|-----------------------------|-----------------------------------------------------------------------------------------------------------------|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------|-------------------------------------------------------------------------------------------------|---------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------|----------------|--|



# AMIGA HAS MULTI-TASKING, DISCOVERY SOFTWARE USES IT!

## FROM NOW ON YOU CAN PRINT OR SAVE ANY SCREEN, FROM ANY PROGRAM, ANY TIME!

GRABBIT takes WYSIWYG\* to the limit.

With GRABBIT you capture exactly what you see on your screen in an instant, regardless of what other programs you're running. GRABBIT works with all AMIGA video modes, including "Hold-and-Modify". It even lets you capture images from animated programs, like the bouncing ball in Boing! What's more, GRABBIT runs completely in the background — transparent to your other software. GRABBIT is always ready for you to use, even while you're in the middle of another program. As if that's not enough, GRABBIT requires only about 10K of your precious RAM to operate, and it supports dozens of printers. It's not a game, it's not a toy, GRABBIT is truly a productivity power tool for your AMIGA!

We believe powerful software should be easy to use. GRABBIT is one of the **EASIEST** programs you'll ever use! Every GRABBIT operation is triggered by one of the "HotKeys", a set of easy-to-remember key sequences that only take minutes to learn. Each HotKey is generated simply by holding down the "Control" and "Alt" keys and pressing one of the designated letter keys. What could be easier?

You won't grow old waiting for GRABBIT to finish printing, either. When we say multi-tasking, we mean it. GRABBIT has a unique TPM (Task Priority Monitor) module which makes sure your other software can still run even while GRABBIT is printing. The TPM module constantly tracks GRABBIT's printing priority, making sure it is neither too high nor too low, but always just right! GRABBIT adds a new dimension to the AMIGA's multi-tasking capability.

GRABBIT supports dozens of different printers because it uses the standard Amiga device drivers. Any printer you can choose in "Preferences" is automatically supported by GRABBIT. You'll get the most from color printers too, because GRABBIT supports full-color printing. In fact, we have seen amazing color printouts produced by GRABBIT on the Oki-Mate 20, a color printer costing less than \$200.00.

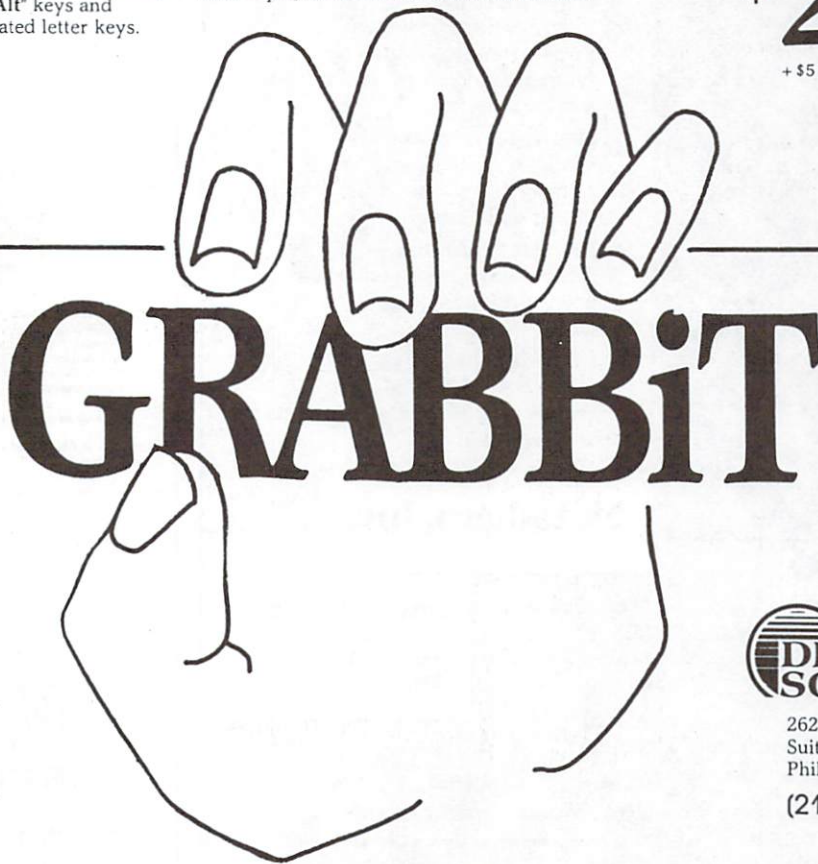
Of course, GRABBIT's abilities are not limited merely to printing; GRABBIT is equally adept at saving screen images to disk — yes, even HAM screens! All GRABBIT

disk files are saved in the popular IFF format, the emerging graphics standard for AMIGA. You can capture any screen to disk for slide-show presentations or later enhancement with any popular AMIGA graphics editor like AEGIS Images or Deluxe Paint. We even include a specially modified PD utility called "SEE", which allows you to view IFF image files quickly and easily. GRABBIT's disk operations are lightning fast because GRABBIT is written in a hybrid of highly optimized C and 68000 Assembler.

Once you start using GRABBIT you'll want it on every disk. You can easily install GRABBIT in your system startup-sequence, so it will always be there when you need it. With all its features this would be a great package at any price. But we think you'll agree with us that GRABBIT's most outstanding feature is **VALUE!** You get all the power of this sizzling new software for an unbelievably low

# \$29<sup>95</sup>

+ \$5 Shipping & Handling



# GRABBIT

**DISCOVERY  
SOFTWARE**

262 South 15th Street  
Suite 400  
Philadelphia, PA 19102  
(215) 546-1533

\*WHAT YOU SEE IS WHAT YOU GET  
Amiga is a registered trademark of  
Commodore Business Machines.  
Images is a registered trademark of  
AEGIS Development Corp.  
Deluxe Paint is a registered trademark  
of Electronic Arts.  
Oki-Mate 20 is a registered trademark of  
Okidata of America.



## MetaScope: The Debugger

MetaScope gives you everything you've always wanted in an application program debugger:

- **Memory Windows**  
Move through memory, display data or disassembled code, freeze to preserve display and allow restoration.
- **Other Windows**  
Status windows show register contents and program state with freeze and restore; symbol, hunk, and breakpoint windows list current definitions.
- **Execution Control**  
Breakpoints with repetition counts and conditional expressions; trace for all instructions or subroutine level, both single-step and continuous execution.
- **Full Symbolic Capability**  
Read symbols from files, define new ones, use anywhere.

- **Powerful Expression Evaluation**  
Use extended operator set including relationals, all assembler number formats.
- **Direct to Memory Assembler**  
Enter instruction statements for direct conversion to code in memory.
- **and More!**  
Log file for operations and displays, modify/search/fill memory, etc.

## MetaScribe: The Editor

MetaScribe has the features you need in a program editor:

- **Full Mouse Support**  
Use for text selection, command menus, scrolling — or use key equivalents when more convenient.
- **Multiple Undo**  
Undo all commands, one at a time, to level limited only by available memory.
- **Sophisticated Search/Replace**  
Regular expressions, forward/backward, full file or marked block.
- **Multiple Windows**  
Work with different files or different portions of the same file at one time.
- **Keystroke Macros**  
Record keystroke sequences or predefine, assign to keys you choose.
- **and More!**  
Copy between files, block copy/move/delete, set tabs and margins, etc.

## MetaTools I

A comprehensive set of tools to aid your programming (full source included):

- **MetaMake**  
Program maintenance utility.
- **Grep**  
Sophisticated pattern matching utility.
- **Diff**  
Source file compare.
- **Filter**  
Text file filter.
- **Comp**  
Simple file compare.
- **Dump**  
File dump utility.
- **MetaSend**  
Amiga to PC file transfer.
- **MetaRecv**  
PC to Amiga file transfer.

Metadigm products are designed to fully utilize the capabilities of the Amiga™ in helping you develop your programs. If you're programming the Amiga, you can't afford to be without them.

Dealer Inquiries Welcome

## Metadigm, Inc.

MetaScope  
\$95.00  
MetaScribe  
\$85.00  
MetaTools  
\$69.95

(California residents +6%).  
Visa/MasterCard accepted.

Amiga is a trademark of Commodore-Amiga Inc.

19762 MacArthur Blvd.  
Suite 300  
Irvine, CA 92715  
(714) 955-2555

## PUBLIC DOMAIN SOFTWARE

PIM PUBLICATIONS, Inc. has  
AMICUS Disks 1 through 8  
and  
Fred Fish Disks 1 through 24  
available at special prices

Per Disk

\$6.00 to Amazing Computing Subscribers  
or  
\$7.00 to non subscribers  
(MA Residents add 5% sales tax)

Make checks payable to:

PIM Publications, Inc.  
P.O. Box 869  
Fall River, MA. 02722

Everyone is encouraged to distribute this software  
freely to your friends, members, and customers.

Please allow 4 weeks for delivery

## Amazing Computing

Publisher: Joyce Hicks  
Circulation Manager: Doris Gamble  
Assistant to the Publisher: Robert James Hicks  
Corporate Advisor: Robert H. Gamble

Managing Editor: Don Hicks  
Hardware Editor: Ernest P. Viveiros Sr.  
AMCUS Editor: John Foust

Production Assistants:  
Ernest P. Viveiros Jr.  
John David Fastino

### Amazing Authors:

Ervin Bobo  
John Foust  
Don Hicks  
Kelly Kauffman  
Perry Kivolowitz  
George Musser Jr.  
Rick Wirth  
Daniel Zigmund  
&  
The Amigo

Special Thanks to:  
Robert H. Bergwall  
RESCO, Inc.  
E.P.V. Consulting  
New England Technical Services  
Interactive Tutorials Inc

Advertising Sales  
1-617-679-3109

Amazing Computing™ (ISSN 0886-9480) is published by  
PIM Publications, Inc., P.O. Box 869, Fall River, MA. 02722.  
Subscriptions: In the U.S., 12 issues for \$24.00; Canada  
and Mexico, \$30.00; Overseas, \$35.00. Printed in the  
U.S.A. Copyright © 1986 by PIM Publications, Inc. All  
rights reserved.

First Class or Air Mail rates available upon request.

PIM Publications maintains the right to refuse any  
advertising.

## Subscription Problems? Moving?

Please let us know

send information to:  
PIM Publications, Inc.  
P.O. Box 869  
Fall River, MA. 02722

If we don't know where you are,  
we can not send you your magazine!



# Amazing Computing

## Table of Contents

July 1986

|                                                                                                         |    |
|---------------------------------------------------------------------------------------------------------|----|
| Temple of Apshai Trilogy review.....                                                                    | 9  |
| Stephen Pietrowicz on the EPYX package                                                                  |    |
| The Halley Project: A Mission In Our Solar System.....                                                  | 11 |
| Stephen Pietrowicz on this addition from Mindscape                                                      |    |
| Flow: a review.....                                                                                     | 13 |
| Erv Bobo looks at this first idea processor for the Amiga                                               |    |
| Textcraft Plus a first look.....                                                                        | 15 |
| Joe Lowery inspects this first update still in beta version                                             |    |
| How To Start Your Own Amiga User Group.....                                                             | 17 |
| William Simpson details the steps to a good user group.                                                 |    |
| Amiga User Groups.....                                                                                  | 21 |
| A list of some of the active Amiga User Groups.                                                         |    |
| AmigaBASIC Tutorial.....                                                                                | 23 |
| Kelly Kauffman on Amiga Basic and math functions                                                        |    |
| Mailing List.....                                                                                       | 25 |
| Kelly Kauffman has created a simple mailing list program to enter and to learn by.                      |    |
| Pointer Image Editor.....                                                                               | 29 |
| Stephen Pietrowicz has produced a little pointer magic with C                                           |    |
| Forth! .....                                                                                            | 41 |
| John Bryan starts the design for a forth application                                                    |    |
| Roomers.....                                                                                            | 45 |
| The Amigo looks inside Commodore at the next generation Amigas under development                        |    |
| SCRIMPER, Screen Image Printer.....                                                                     | 49 |
| Perry Kivolowitz completes his explanation of the Scrimper program.                                     |    |
| The Amazing C Tutorial.....                                                                             | 57 |
| Busy Stephen Pietrowicz takes the podium temporarily for<br>John Foust to explore structures and unions |    |

### The AMICUS Network:

|                                                                                                       |    |
|-------------------------------------------------------------------------------------------------------|----|
| Amicus.....                                                                                           | 67 |
| John Foust offers his experience with telecommunications (which is considerable)                      |    |
| Fun with the Amiga Disk Controller by Dr. Thom Sterling.....                                          | 73 |
| The AMICUS Public Domain Software Library: AMICUS Disks 1 to 10 are listed and detailed .....         | 75 |
| Fred Fish Disks, 1 through 25: a special list of the current Fred Fish Library of Amiga Programs..... | 77 |

### Departments:

|                           |    |
|---------------------------|----|
| From the Editor.....      | 4  |
| Letters.....              | 6  |
| Index of Advertisers..... | 80 |



# From the Editor:

by Don Hicks

## User Groups

We have all been there.....

You decided you would make the plunge and buy the Amiga, but your spouse is less than pleased with your decision on another "toy". As you mount the reasons this purchase is so crucial, you know there is one more reference you should check.....

It is Seven PM on Saturday and you just picked up your new prized Amiga and have put all the pieces where you think they should fit. Unfortunately, when you turn on the power, nothing happens.....

The new software package, which looked so good at the dealers, came with a 200 page manual and a degree in technical documentation if you are able to read it.....

For the third time, you have entered data in your everyday spreadsheet, and it has completely ignored your efforts. the computer is completely "locked up". A silent blank screen sits and watches you.....

These, or many other situations just as painfully familiar, almost always lead to one thing, ask a friend.

It is not easy buying a computer for the first time and I am positive almost everyone out there searched for a compatriot to provide moral support through the purchase. This is the very heart of User Groups.

I belonged to a user group for about six months before I had any idea we were one. Back in the "old days", Ernest Viveiros and I car pooled to work. We talked about different things on the drive, until one day he mentioned he had an Apple II (not "plus"). From then on, things quickly changed.

I started asking questions such as why he got it, what good was it, what did he do with it. The questions slowly turned to, will it do this, what software will do this, how much are they.

Ernest was (and still is) a good friend, he put up with my questions and even went with me to purchase my first machine. He even took the time to help me set it up that afternoon, Superbowl Sunday.

This is a user group. It can be as small as two dedicated friends finding their way through a maze of problems or as large as thousands of people sharing ideas. If it serves the needs of the members, it is a user group.

In no other facet of American technology is there a need for people to gather together and discuss the pros and cons of brands, or to help individuals understand how to use their machines, or even produce and distribute programs to make their machines work.

The computer is a highly technical piece of equipment that in most cases has been purchased with considerable thought and budget juggling. Support through dealers is one thing, but a friend who has experience or at least wants to learn with you, is invaluable.

I have been a member of several user groups, some national and some local. Some are better than others, a user group is only as good as the members can make it.

In a national user group, there is probably a "home chapter" in a large city with many members, where meetings are held. However, they normally produce an extremely good newsletter and have a large volume of Public Domain software to distribute. This access may be especially welcomed by individuals in areas where there are no common interest groups. Often, it is worth the price of membership just to have access to this material.

In a local group, you must take a positive active part in the working of the club. Either as an officer or as a "working member" your contributions are needed.

In a local group, meetings are extremely important. There is always some new bit of news that is interesting. There is often a demo of a member's or developer's software. But, most important, it is a place to share ideas, develop new applications for your machine and make friends.

Because we believe strongly in the need of user groups to promote computing and especially the Amiga, we have dedicated this issue to these groups.

We have included an article on how to develop a user group, a listing of several groups around the country and articles written by members of user groups.

It is our hope that you already belong to a group or soon will. After all, Amazing Computing is produced by a group of users dedicated to getting the word out on the Amiga.

We believe that a good user group is the best investment you can make for your Amiga.



Don Hicks  
Managing Editor



# New Amiga Products From The Developers of Amiga C.

**Amiga C Compiler**—\$149.95  
Everything you need to develop programs on the Amiga, including a full set of libraries, header files, an object module disassembler, and sample C programs.

**Unicalc**—\$79.95 A complete spreadsheet package for Amiga, with the powerful features made popular by programs such as VisiCalc, SuperCalc, and Lotus 1-2-3. Unicalc provides many display options and generates printed reports in a variety of formats and print image files. Supports 8192 rows of 256 columns, and includes complete on-line help.

**Lattice MacLibrary**—\$100.00  
The Lattice MacLibrary is a collection of more than sixty C functions enabling you to rapidly convert your Macintosh programs to run on the Amiga. This allows you to quickly and efficiently take advantage of the powerful capabilities of the Amiga.

**Lattice Make Utility**—\$125.00  
Automated product generation utility for Amiga, similar to UNIX Make, LMK rebuilds complex programs with a single command. Specify the relationships of the pieces, and automatically rebuild your system the same way every time.



## Lattice

Phone (312) 858-7950 TWX 910-291-2190

INTERNATIONAL SALES OFFICES:

Benelux: De Vooght. Phone (32)-2-720-91-28. England: Roundhill. Phone (0672) 54675

Japan: Lifeboat Inc. Phone (03) 293-4711 France: SFL. Phone (1) 46-66-11-55

**Text Utilities**—\$75.00 Eight software tools for managing text files. *GREP* searches for specified character strings; *DIFF* compares files; *EXTRACT* creates a list of files to be extracted from the current directory; *BUILD* creates new files from a batch list; *WC* displays a character count and a checksum of a specified file; *ED* is a line editor which utilizes output from other Text Utilities; *SPLAT* is a search and replace function; and *FILES* lists, copies, erases or removes files or entire directory structures.

**Lattice Screen Editor (LSE)**—\$100.00 Fast, flexible and easy to learn editor designed specifically for programmers. LSE's multi-window environment provides the editor functions such as block moves, pattern searches, and "cut and paste". Plus programmer features such as an error tracking mode and three assembly language input modes.

---

## OTHER AMIGA PRODUCTS AVAILABLE FROM LATTICE:

**Panel:** Screen Layout Utilities—\$195.00

**Cross Compiler:**

MS-DOS to Amiga C—\$250.00

**dBC III:**

library of data base functions—\$150.00

**Cross Reference Generator**—\$45.00

With Lattice products you get *Lattice Service* including telephone support, notice of new products and enhancements, and a money-back guarantee. Corporate license agreements available.



# Amazing Mail

Dear Editor:

I work at a company that deals with a number of kinds of computers (including our own 11/73 and 68000 real-time machines), and employs a number of professional programmers of differing abilities.

Recently, a document was circulated at work describing a (VERY) simple C prime number algorithm, and the results of using it to benchmark six computers including the Amiga. I have copied it below. As you can see, it doesn't make the Amiga look so good.

These results, of course, are misleading: what is actually benchmarked is the computer and its C compiler.

On the other hand, at least two of the other machines - the IBM PC and the HP 150 (an 8086-based semi-PC-clone) -- also used Lattice C compilers.

Knowing that the Amiga couldn't be THAT bad, I disassembled the Amiga Lattice code for the algorithm. This revealed a basic Lattice (3.03) weakness: data is constantly moved from registers to memory and back, and from register to register, for no good reason. In addition, however, the modulo operator ('%') makes use of a Lattice subroutine named '\_CXD33'.

On a hunch, I replaced Lattice's routine with my own version:

```

**      import                      export
**      d1 = divisor                = result
**      d0 = that which is divided  = remainder

xdef _CXD33      tell the linker we are here

_CXD33 tst.l d1      divide by 0 (tsk tsk) ?
    beq  LEEV      leave with register values alone
    divs d1,d0      low word = result;
                    high word = rmndr

    move.l d0,d1     copy results
    ext.l d0         make result 32 bits
    swap d1          put remainder in low word
    ext.l d1         make remainder 32 bits
    LEEV  rts        take us back...
```

The hunch proved correct! As you can see, in the "revised" version, the Amiga comes in second and is

actually catching up on the leader (the HP 150) in the later rounds.

Sincerely yours,  
Dr. Gerald Hull

```

#include "stdio.h"

#define MAX 4020

main()
{
    int i, n = 0;

    while (++n <= MAX)
    {
        i = 1;
        while (++i < n)
            if (n % i == 0)
                break;
        if (i == n)
            printf("%d\n", n);
    }
}
```

| System      | >1000 | >2000 | >3000 | >4000 |
|-------------|-------|-------|-------|-------|
| 1 HP 150    | 10    | 26    | 46    | 72    |
| 2 IBM PC    | 13    | 32    | 62    | 99    |
| 3 ATARI ST  | 14    | 35    | 75    | 175   |
| 4 AMIGA     | 29    | 92    | 188   | 313   |
| 5 ATARI 800 | 215   | 748   | --    | --    |
| 6 C-64      | 305   | 1039  | --    | --    |

Amiga times with my revisions:

|         |    |    |    |    |
|---------|----|----|----|----|
| 7 AMIGA | 12 | 28 | 48 | 73 |
|---------|----|----|----|----|

*Thank you Mr. Hull, it is nice to see what a little patience can prove.*



Dear Amazing Computing:

I'm very happy to see you, and hope to get to know you better. I also have a few problems which I'd appreciate your help in addressing. I bought my Amiga because I needed something heftier than my Apple II for working on a book. Correction: I bought a new computer for that reason. I chose the Amiga because I was helpless to resist it. My first problem is that I have many textfiles on the Apple that I need to transfer to the Amiga, and have hardly a notion how to begin. I think I can get them out of the Apple with a simple basic program to send a character at a time through my serial interface card, but how do I get them safely tucked away in a textfile in the Amiga? Any suggestions?

To the same end (of getting my book done), I'd be oh so grateful for a good look at available and soon to be released word

processors. I had planned to get Enable/Write, but as its release seems to get farther away instead of closer, I may have to settle for something else. Has anybody seen enough of it and other word processors to get a comprehensive review together yet?

Thanks for being there.  
David C. Thomson  
Versailles, KY

*To answer your first problem, most terminal programs will handle the transfer of data from one machine to another through a "null modem cable" connected to the serial ports. You will need a program for each machine. You may require a stripping program to provide necessary carriage returns and line feeds.*

*In regards to word processing, We have included a review of Textcraft plus in this issue and are planning a*

*review of scribble! soon. As to Enablewrite, we have not seen the product so far....*

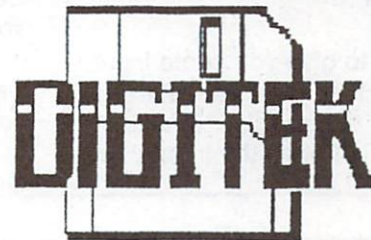
I want Amazing Computing. My wife wants me to support industry in her hometown (Fall River). Please start my subscription.

James H. Wilson  
Madison, WI

Dear Sirs,  
You truly are producing an amazing publication, and performing a great service for Amiga owners and developers. I have had my Amiga since late September, purchasing about the 4th bought in the city of Gainesville, Florida. I was initially somewhat frustrated, having no documentation for such essentials as the CLI and ED. I was THRILLED to get my hands on a copy of your magazine.

## deluxe clip art #1

- Deluxe Paint and Aegis Images compatible.
- Deluxe Print compatible.
- Many images to choose from — Cars, Animals, Borders, and more. . .



SOFTWARE DEVELOPMENT®

2519 Bonita Drive  
Highland, Ca. 92346  
(714) 864-2846

\$29.95

### Be the artist you always wanted to be!

Please add \$2.00 for shipping and handling.

Deluxe Paint, Deluxe Print is a trademark of Electronic Arts.  
Aegis Images is a trademark of Aegis Software.



I am thoroughly impressed with the power of the Amiga and plan to implement it in my research at the University of Florida college of Medicine, Department of Anesthesiology for data collection and analysis, graphics development, and telecommunication (not to mention such mundane tasks as word processing). I am tremendously excited regarding the potential of the Amiga to function in these and other tasks.

I am happy to support your publication, and look forward to many years of mutual benefit.

Sincerely,  
Raymond H. Castenholz, M.D.  
Gainesville, Florida

Dear Friends,  
I have read all four issues from cover to cover and believe that your publication is the tool I need to make the most use of the Amiga I have purchased. Since my dealer is a 45 minute drive away and your magazine is a vital source to my computing efforts I do not want to miss an issue.

I would like to give you some input as to what I feel helps educate and inform me about this "Amazing" machine. First of all the tutorial on

C has been great and I would like to see even more information on style and form. Next, is the review section. I hope you will include reviews on utilities and other languages and not just entertainment software. Next, the "Roomers" section of the magazine is very helpful in keeping spirits high on the development of extras for the Amiga. It might be nice if there was a way for you to indicate to us how much faith you had in each "roomer", say like a percentage indicator. Last, but surely not least are the programs that have been included in the issues. They have provided insight into the C language.

Sincerely,  
Lynn D. Shumaker  
Phoenix, AZ

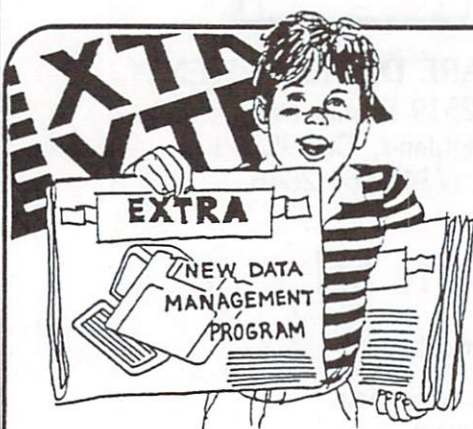
Dear AMAZING,  
How nice to find a publication that is really putting out the kind of nitty-gritty AMIGA information we are thirsting for! The IBM Drive Interface article was great - and so clearly presented it was elegant. Congratulations and good luck with your great new publication.

Sincerely,  
John L. Rehak  
Integrated Systems  
Garden Grove, CA

Dear Amazing Computing,  
At last, my quest has ended. I recently visited relatives in the United States, and popped into a computer dealer somewhere in Michigan. At this shop I saw the most AMAZING Amiga magazine! I mean I have met my main magazine!! This magazine was incredible!!! It should come as no surprise to your now-inflated egos that the magazine was called, oddly enough, AMAZING COMPUTING. Of course, I bought the magazine, which brings us to my problem. You see, such an informative magazine is addictive if you own the most addictive computer in town. But alas, in this town, this AMAZING magazine is not readily available. The only solution to my problem seems to be to bribe you people at PiM Publications into sending me more of these AMAZING magazines.

I hope the bribe worked,  
John Rampelt  
Waterloo, Ontario, CAN

*Thank you all! We will continue to attempt to provide the type of information you want. However, please remember, we are in constant need of your article submissions. We need all the Amazing Writers we can get.*



Eastern Telecom Inc.  
9514 Brimton Drive  
Orlando, FL 32817

## It's Finally Here !! A Database Management System for the AMIGA with all the EXTRAS:

- EXTRA—Easy to use. No new language to learn, just look at the pictures.
- EXTRA—Flexibility, with pull down menus, automatic filing and field sizing.
- EXTRA—On line help, Just press for command information.
- EXTRA—LOW introductory price... **JUST \$49.99 !!!**

We've got ALL the EXTRAS for you.



Order NOW! (305) 657-4355



# Temple of Apshai Trilogy

## Review

by Stephen R. Pietrowicz

CIS [73047,2313]

People Link: DR RITZ

Usenet: ...lihn4!pur-ee!gould!houligan!srp

Many of the games first available for the Amiga were adventure-type games. Most of them were text adventures that had been out for other computers for several years. New adventures that exploit the Amiga's graphics and sound capabilities have appeared during the past few months.

Epyx Inc. recently released an old favorite into the Amiga graphic adventure market, the Temple of Apshai Trilogy. This program combines three separate programs, "Temple of Apshai", "Upper Reaches of Apshai", and "Curse of Ra". All were available for almost four years on many different systems.

This fantasy adventure lets you explore the Apshai area and battle monsters to gain fame and fortune. If you've played Rogue or Hack on a mainframe or minicomputer, you'll be pleasantly surprised at the similarities and the differences.

Along with the program disk, the package includes an instruction manual with over 80 pages and a quick reference card. The manual includes the background of the adventure you're about to take.

In "The One Minute Adventurer" section, you'll get a quick explanation on how to select your character, purchase armor and weapons, and start your adventure. I suggest that you follow the manual's instructions in the "Rules of Apshai", and experiment with the programs pull-down menus. The manual isn't specifically written for the Amiga; it was written for the Apple/64/128/Atari versions.

It was a bit confusing when I clicked the mouse to clear the main title screen and the screen went blank with no explanation. I expected to get a selection menu as the manual stated.

The first pull-down menu has entries to create, enter, save, load, delete and obtain a summary of your characters. First time players should load the predefined character "Brian Nailfoot" from the game disk, especially if you haven't played any adventure-type games before.

Your character has 6 attributes - strength, constitution, dexterity, intelligence, intuition, and ego. Each attribute is rated from 3 to 18, 3 being the worst rating and 18 being the best. These physical and mental attributes all help to determine how you perform in different situations during your adventure.

A higher strength will help you carry more and do greater damage with your weapon against opponents. A low intuition might make you miss that trap in front of you after you've just checked to see if a trap was there. All of the attributes are clearly explained in the manual.

The "create" option will randomly select your attributes and the silver you start with. The "enter" option lets you select your own attribute values, silver, weapons, and armor. By making all your attribute values 18, filling your backpack with arrows, getting a magic sword and magic armor, you can be nearly invincible, but you'll lose the challenge of the game quickly.

When you randomly create your character, you'll start with a limited amount of silver pieces. You'll use the silver to "outfit" your character for battle. The second menu lists the types of armor and weapons you can buy.

For example, the different types of armor you can buy are leather, ringmail, chainmail, partial plate, and full plate. The better the armor, the more expensive it is. Since you've always got to watch your silver pieces, you can even haggle with the innkeeper selling you his wares! If you make an offer that's too low he'll complain that he has children to feed, or that he can't believe the price you want to buy that weapon for! You can also buy healing salves which help heal your wounds when you're adventuring.

The third pull down menu lets you choose which adventure, and which level in that adventure you wish to start. When you start adventuring, movement while inside the temple can be controlled by the mouse or the keyboard. Surprisingly, each is equally easy to use.

Your character moves through each room picking up treasure, searching for traps and secret doors, and battling monsters. Monsters vary in strength and type as you move lower into the dungeon. You have to be very careful to watch your fatigue and wound levels.

If your fatigue level drops to zero or below, you've got to rest; if your wound level drops to zero, you're dead. The more you're carrying, the more fatigued you'll be, so don't be too greedy! Monsters lurk around every corner, and if you collapse from fatigue, they'll show no mercy.

The variety of monsters is quite amusing, especially in the second adventure. You'll do battle with everything from giant rats to killer tomatoes! You can communicate with some monsters, such as drunken sailors. They will allow you to pass without bother.

Battle with monsters is not arcade style. You'll take turns with the monster to decide what to do. You have the options of attacking, thrusting, parrying, shooting your bow, or running away. The bow is quite handy when doing battle with really nasty monsters since you can shoot from a distance, and the monster can't hit you until it is near you.



I really enjoyed the extra touches that were put into the game. When your adventurer moves down the hallway, you can watch him move his arms and legs as he walks. The information window shows a little monster peeking over the window's edge. The reference card will help you remember the game's commands.

The only things I'd like to change in the program are the room descriptions, and the room layouts. When entering a new room, you must either look up the room description in the instruction manual, or from a pull-down menu. It would be more enjoyable if the descriptions were added to the main screen, since you'll tend to go through the dungeon without constantly looking up the room descriptions.

It would also be nice if the rooms were drawn to fit the descriptions. The rooms are plain and empty, except for the treasure or monsters that might be inside each room.

The Temple of Apshai Trilogy requires a one disk, 512K Amiga system. The disk, unfortunately, is copy protected, so you'll have to be careful when you save games to the disk.

**EPYX, Inc.**  
1043 Kiel Court  
Sunnyvale, CA 94089  
Disk \$39.95

•AC•

# DON'T WAIT!

PiM Publications, Inc. wishes to thank all of the fantastic Amiga owners who have accepted Amazing Computing as a resource for their Commodore Amiga. The response was far beyond our expectations. Your letters praised Amazing Computing™ and encouraged us to continue to deliver Commodore Amiga information and programs.

Amazing Computing's success does have one draw back, our Back issues are in short supply. If you have not purchased a copy of the above issues from your Amiga dealer, or he has sold out, PiM Publications will make these back issues available for as long as they last at our Back issue price of \$3.50

Please send to:

**Back Issues**  
**Amazing Computing**  
**PiM Publications, Inc.**  
**P.O. Box 869**  
**Fall River, MA 02722**

If you want to be sure you do not miss a future issue, then subscribe!  
Amazing Computing is available by subscription for 12 issues:

\$24.00 in the United States  
\$30.00 in Canada and Mexico  
\$35.00 Overseas

First class rates available upon request  
Please allow four to six weeks for delivery

**Amazing Computing™** your resource to the Commodore Amiga



# The Halley Project: A Mission In Our Solar System A Review

By **Stephen R. Pietrowicz**

Halley Project is a simulation of solar system travel. You are asked to become a pilot for the Planetary and Lunar Aerospace Navigation and Exploration Team, P.L.A.N.E.T.

The package includes a cassette tape with the briefing for your mission. (Those without a cassette player can read the transcript of the tape, included in the Top Secret mission dossier).

Also included in the package are a small star chart to help you navigate, a quick reference guide, and a booklet with coupons for several astronomy related books, magazines and posters.

Unfortunately, the disk that is included with the package is copy-protected. It would be nice to be able to make a backup copy of the program, especially since the disk must be left writable to let the program record your times for successful missions.

Many companies have hastily been porting software from other machines to the Amiga in hopes that software starved owners will buy it up as soon as it hits the shelves. Unfortunately many of these ports don't take advantage of the Amiga's powerful capabilities. Mindscape decided instead to rewrite "The Halley Project" to use Amiga's sound and graphics with impressive results.

After a quick browse through the rest of the dossier, you are ready to begin your mission.

Make sure you have your sound turned up for the opening sequence. You'll be impressed with the digitized music!

Once you have typed your name for the mission records you don't have to use the keyboard again. You use the mouse to control the functions of your ship.

You begin each mission by flying away from Halley's Comet. Instructions for each of your missions are flashed up on the screen as you complete each leg of the flight. You attain new rankings each time you successfully complete a new mission.

In your first mission ranked as "Raven", you must navigate to and land on Earth. As you rise in the ranks, you must navigate to a greater number of destinations to complete each mission.

Clues also become slightly more cryptic in some cases. Two screens are available to help you navigate from the comet to your mission objectives: the control panel, and the radar screen.

On the control panel screen, a small switch below the viewing screen allows viewing only in a horizontal plane. A control cluster allows you to choose which direction you fire your thrusters to put your ship into motion.

Another control lever sets the power level of the thrust adjustments. The high setting allows for quick acceleration when you are far away from the planet or moon you wish to land on, and the low setting is used to navigate when you are ready to land.

The astrological constellations along with the direction indicator give you a reference point to your mission objective after locating it using the radar screen.

The radar screen helps you orient your ship to locate your mission goals. The names of the constellations on your star chart circle the screen and are used as reference points.

The zoom in and zoom out gadgets let you adjust the radar to put the planet you wish to fly to on the outer border of the screen. The legend at the bottom of the screen adjusts as you zoom in and out, and gives the distance from your ship to the outer edge of the screen in millions of kilometers.

Once you have computed the distance to your goal, and have noted which constellation is close to it, you are ready to fly to your destination.

You use the viewing screen gadget to scroll to the constellation you noted on the radar screen. Since the distance you have to travel is very large, your ship is equipped with "hyperspace" capabilities. Increasing your ship's speed to 300,000 KM/second puts it into hyperspace mode.

While in hyperspace mode, the distance you are covering is shown on the screen. When you get close to how far you want to travel, pressing either mouse button will stop your ship.

Now use the radar screen to re-orient yourself again, and fly closer to the planet. When you get close enough to the planet, you will be able to see it on the viewing screen, and the distance to the planet will be shown.

When you get close to a planet or moon, you have to find a landing site. The landing site will be automatically locked in once you pass over it while you are in orbit. The ship must be less than 100,000 KM to land. The theme music begins playing to remind you that you are closer than 100,000 KM from the planet.

You will be able to find the landing site more easily if you use the viewing gadget to sweep back and forth on the planets surface. When the landing site is found, the locator will flash and beep. Clicking on the automatic landing system will land your ship.



# COLONY SOFTWARE



931 W 21 ST  
NORFOLK VA  
23517

(804)625-2089

BBS  
(804)625-1945

## FileCraft!

Data Base Manager - \$79.95  
Easy-To-Use, w/4 Ready-To-Use  
Data Bases. Pop-In the disk -  
and START USING THE PROGRAM -  
IMMEDIATELY!

Powerful, Fast, and Tested..  
Programmable- Data Files  
addressed, outside DBM.

Great Documentation - BUT  
YOU WON'T NEED IT.

Includes LABEL-MAKER!

## MailCraft!

Mail Merge - \$25 -

Use w/FileCraft! (above) to  
merge letters created with TEXT-  
CRAFT and data base names/  
addresses.

Makes personalized form  
letters - EASILY.

## CashCraft!

Point-of-Sale - \$99.95 -

Cash register/Inventory Sys.

ALL OTHER COMMERCIAL AMIGA  
SOFTWARE. Approximately 20% OFF!

Lowest Prices - Write/Call  
for Catalog/Hints - \$5 -.

To Be Released (Real Soon Now):  
SLIDE SHOW CONSTRUCTION SET -\$25  
MARION-THE-LIBRARIAN - \$40 -.

## RENTAL SOFTWARE

Access our DISK LIBRARIES for  
AS LITTLE AS \$2/DISK - RENTAL:

1000+ Disks of IBM-PC software  
converted to 3-1/2" disks. (Use  
w/Transformer software, w/o 5"drive.  
100+ disks of AMIGA software.

Now you will receive your instructions of where to fly next. Each mission ends by flying back to Halley's Comet. (Halley's Comet is always shown on the radar screen as a blinking dot so you can distinguish it from planets). If you accidentally crash into the planet you are trying to land on, you receive a 5 minute penalty which is added to your score.

Planets and moons are shown as different colored circles on your viewing screen, and do not have any details on them to make the 3D transformations easier. For example, Earth is completely blue. After landing on any of the planets or moons, your viewing screen shows a scene you might see if you really did land there. Some are quite impressive!

I do not particularly like the hyperspace mechanism that Halley uses. I found it a bit boring having to wait for my speed to increase back up to 300,000 KM everytime I dropped out of hyperspace. Other than that, flying the ship is quite easy since you control everything on the ship with the mouse.

There is even a mission description gadget that will tell you what your mission is in case you did not make note of it when you landed. (This has come in handy a few times already!)

Trying to go completely out of the solar system results in termination of your mission, and you are sent back to the comet. I discovered this after I completed the last leg of the Falcon mission. It seemed a bit harsh at the time, but I'll never try it again!

The Halley Project claims to be a real-time simulation, and also claims to have accurate model gravity. Neither of these claims is true.

A timer on the viewing screen keeps track of how long you take to complete each mission, but reducing your speed to zero stops the clock and lets you orient yourself without any time penalties.

As a test of gravity, I tried to fly as close as I could to the sun. I stopped about 1000 KM above the sun, and waited to see if it's gravity would pull my ship in. It didn't. As a further test, I set the speed control on low and moved in closer. I was able to stop at 0 KM away from the sun and nothing happened!

Despite these inconsistencies, Halley Project is definitely worth buying, especially for children. It is entertaining while at the same time educational. It should provide hours of entertainment for you and your whole family.

"The Halley Project: A Mission In Our Solar System"  
runs on an Amiga system with 256K, one disk drive, mouse,  
and KickStart 1.1.

It has a list price of \$44.95.

It is available from:  
**Mindscape, Inc.,**  
3444 Dundee Rd.,  
Northbrook, IL 60062.

•AC•



# FLOW:

Reviewed by Erv Bobo

Subtitled an idea processor for the Amiga, FLOW is just a little more than that, having capabilities of creating outlines, keeping an appointment schedule, even serving as a filing system.

Although the on-screen action resembles ThinkTank, a similar program for IBM and Macintosh, FLOW seems easier and faster to use because of the implementation of the Intuition interface - mouse and pull-down menus. And, because it resides on a Workbench disk, there should be no problems in making a working copy or in installing it on a hard disk.

To show you exactly what FLOW can do, several examples are included in the drawer labelled Examples. Click it open and you will see a file labelled "Amiga Magazines". In opening this file, you'll see two entries "AmigaWorld" and "Amazing Computing". Guess which one we're going to try.

With the mouse, you put the I-beam-shaped cursor next to the AMAZING listing, then go to the SubMenu, pull it down and click on Expand. Now the file opens to show you a listing of every issue of AMAZING - all two of them, as of the time FLOW came to market. Clicking on either of these entries and clicking Expand will open that file further to show you the contents of that issue.

Had we, at the outset, chosen Expand All, every entry in the AMAZING file would have been opened with one mouseclick. Going in reverse, we can close all or part of the file in much the same way: with the cursor at a subheading, click on Collapse and the entries under that heading close up. Click on the main heading and choose Collapse and the entire file closes

to show just the one entry - "Amazing Computing"

So what? You have both issues right here and you can open either one and see the table of contents. As for outlines, you learned how to do that in school and, even though today you simply jot things down on scraps of paper, it's easier than turning on the computer, booting this and that, making entries and saving them and....

In the first place, FLOW gives you a Find feature. If you want to quickly locate, say, the Super Spheres listing, select Find and when the requester box appears type in a keyword from the title. Quicker than you can open issue one, FLOW highlights the entry for you providing it is a heading or subheading.

In the second instance, your school outlines were done on notebook paper and, if you did them well, much erasing and rewriting was involved. With the Cut, Paste and Copy features of FLOW, it is rather like having each line of your outline on a separate strip of paper, to be moved, arranged and rearranged until you have it right.

To create such a file, type in your title. For the next entry, a subheading, tab once. As soon as you tab, FLOW recognizes your first entry as a heading and converts it to bold type. Enter your subheading and on the next line, for the body of your outline, tab twice. Again, the tabbing operation tells FLOW the last entry was a subheading and it changes to bold type. The result is a neat, clean form, suitably indented and with all headings in bold, making the later reading or scanning quick and easy.

With other options from the pull-down menus, you can change headings; cut, copy and paste; replace or erase headings and subheadings; and alphabetically sort entries in either ascending or descending order.

Although the term "what-if" is usually reserved for the multiple models that can be created with a spreadsheet, it is just as descriptive of what you can do with FLOW. Jot down your ideas as they come to you and later experiment by changing the order of events or entries until you have the one sequence that is exactly right for your project.

And speaking of projects, though FLOW uses text rather than text and symbols and though it uses a vertical list form instead of horizontal, there is probably no reason why it could not be used effectively to set a sequence of the timing of events usually called Critical Path Analysis.

Documentation for FLOW is sparse, about forty-eight pages, but you should be up-and-running after following the initial walkthrough in the first chapter. After that, the possibilities of the program will only be limited by your imagination.

ThinkTank was a successful program because it filled a need and FLOW should emulate that success because it fills a similar need for the Amiga. Although, as I did, you may not see that until you've worked with the program. Once you do, you'll find it to be indispensable.

New Horizons Software  
P.O. Box 43167  
Austin, Texas 78745  
Disk \$99.95



# DISKWIK

## AMIGA DISK UTILITY

### The First Disk Repair Kit Available

- : Restore deleted files
- : Repair errors
- : Recover data from damaged files
- : Edit in HEX or ASCII
- : Simple control using mouse
- : Complete Documentation describing the disk and all the sector types

Diskwik gives you total access and control of your disks far beyond that which DOS allows. DISKWIK has features for the novice and advanced user, including such life saving features as restoring deleted files, and eliminating any errors on the disk except those due to defective disks. The INFO option will give you block type, filename, bytes in file, protection status, and more...

The documentation included is worth the price alone. In it you will learn how data is stored, how the directory works, and all the different types of sectors are explained. Diagrams matching the screen on your monitor make it easy to locate and identify the different sections. All this and more gives you the knowledge to fix corrupted disks, or just experiment.

Advanced features include editing in HEX or ASCII, copy blocks to the same or another disk. Re-format tracks, correct checksums, open a file to a ram disk to save one or more individual blocks for later use.

DISKWIK fully utilizes the mouse to make moving around the disk as easy as the push of a button. All this for only \$49.95 makes it a true value. DISKWIK is available from:

# *Tigress*

P.O. Box 665  
Glendora, CA 91740  
(818) 334-0709

**Dealer Inquires invited**



# TEXTCRAFT PLUS

## A (Pre)REVIEW

by Joseph Lowery  
(AMuse, New York Amiga User's Group)

If this isn't a healthy sign, I don't know what is. Imagine a company whose product is often bundled with the hottest new computer. A pretty safe, comfortable spot, right? No reason to put yourself out. Just sit back and rake it in. What you'd expect, right?

Well, take a look at TextCraft Plus. The folks at Arktronics (TextCraft Plus programmers Mike Rivera, Ralph Stefan, and Eric Thomassian) have taken an acceptable product, cut the cuteness and gimmickry by about 85%, doubled the productivity possible and broken new ground as well! Plus, they have maintained the only decent connotation of an "entry-level word-processor", and that's ease-of-use.

TextCraft Plus now operates as a task on the Amiga. It appears in a sizeable window, using your Preferences color (either the Workbench or Intuition variety). You can open multiple windows of TextCraft Plus and cut and paste between them. TextCraft Plus can even print from one window while you edit the same document in another (although this slows printing considerably).

The other major shift in TextCraft Plus is in its handling. A menu item entitled "Select" has been added which allows you to quickly highlight everything from a single character to the entire document. Then you can act upon what you have selected - and do anything TextCraft Plus can do: cut, copy, erase, make bold, underline, single or double space, flush left, flush right, justify - the whole gamut. This concept is referred to as the "noun/verb user interface," as in "Sentence - Italicize" or "Page - Erase."

Also, you no longer need to change gears before you select. Previously you had to switch to the style brush, choosing an option such as underline, and highlighting your text with the style brush and then return to the editing mode. Now, with TextCraft Plus, highlight text by dragging the mouse (or choosing an option from the drop-down Select menu, or use one of the first five function keys) and choose your "verb." You return to editing with one click of the mouse.

The original TextCraft's main appeal for me was its WYSIWYG capability ("What You See Is What You Get" - I love to pronounce it "Wizzywig"). The "Print Previews" common to many other word processor's make it difficult, if not impossible, to do certain tasks such as charts and number tables.

TextCraft Plus has taken this one step further and will allow you to toggle the headers and/or footers into view. (Incidentally, both open their own window so that you can have more than one line of text in each.)

TextCraft Plus is claiming support for IFF text files now, and in a future release will be able to intermix your Images or DeluxePaint pictures with your text. Now that's wizzywig!

Other notable improvements include vastly improved directory handling, form saving and loading, better cursor movement through the document, auto-saving of back-up, merging of text files, and page breaks.

A major enhancement is the inclusion of a mailmerge feature for the printing of form letters and the like. TextCraft Plus uses a very simple procedure for producing the merge list and the main document. At the top of the merge list type the key words, like this:

"First Name, Last Name, Address."

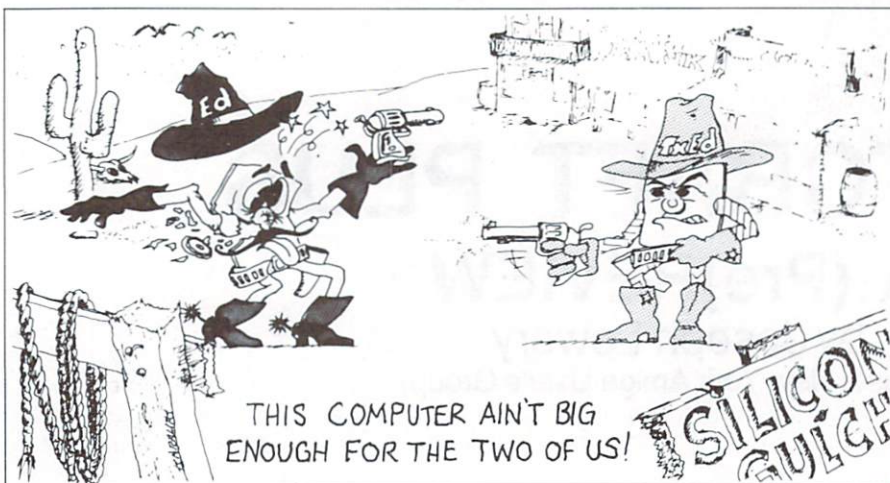
Then skip a line and start your list:

"Joe, Schmo, 322 Kokomo."

Save this as a "Merge List" and then draft your main document. To use the information in your Merge List, you use double brackets around your key words:

"Dear <<Last Name>>,  
Can I call you <<First Name>>? As a happy homeowner at  
<<Address>>..."





# FIRE YOUR EDITOR.

## And put Microsmiths' TxEd to use for you.

- ★ FAST display updates - more than TWENTY TIMES as fast as "Ed".
- ★ Designed from the ground up to take advantage of the Amiga user interface.
- ★ Multiple windows.
- ★ Very easy to learn, use menus for online help.
- ★ Simple, elegant set of commands.
- ★ Alternate command keys shown in menus allow fast command entry for experienced users.
- ★ Compact code works well with Amiga's multi-tasking.
- ★ The first Amiga directory requester that doesn't make you wait.
- ★ All around utility editor is good for programmers, and also useful as a simple word processor. Great for use with terminal programs.
- ★ Pronounced "Tex Ed" as in "Tex Ed, the Faster Editor in Silicon Gulch."

To order, send \$59.95 in check or money order plus \$2.50 postage and handling to: Microsmiths' TxEd, P.O. Box 561, Cambridge, MA 02140. Tel.: (617) 576-2878. Mass. residents add 5% sales tax. Amiga is a trademark of Commodore-Amiga, Inc. Designed by C. Heath. Dealer inquiries invited.



**MICROSMITHS, INC.**

P.O. Box 561, Cambridge, MA 02140

To print the final documents select "Print Merge" and follow the prompts. Very simple, very impressive.

### Help is available on three levels.

"Quick Reminder" brings up a screen, similar to the original TextCraft, explaining the symbols residing in the ruler (which you can now hide).

"Keyboard Help" is a scrollable listing of the keyboard equivalents for all the TextCraft Plus commands, including a wide range of cursor movement and text scrolling. One such handy combination is "Control + nnn" where "nnn" is any number on the numeric pad and which will whisk you to the top of that page. I have dreamed of such a feature.

The final help resource is the set of tutorials which now reside as a separate file on the disk. These have been streamlined and "de-cute-ified" while maintaining the key animation sequences. Blessedly, these routines have been speeded up; watching the cursor crawl across the screen never was my idea of a fun way to spend the afternoon.

I mentioned above Arktronics' plans to intermix IFF graphic files and text. They also speak of supporting multiple fonts much as the new NotePad on Workbench 1.2 will. I admire and respect this commitment to one's product and strongly feel that this attitude will go a long way toward convincing the public of the Amiga's staying power.

Release date and up-grade policy of TextCraft Plus have not as yet been set by Commodore-Amiga. However, the most hopeful (ASAP and free, respectively) have not been ruled out. Commodore-Amiga would do well to go by following through with this exciting enhancement. TextCraft Plus is one steep up-grade.

P.S. I have to confess. I kind of miss the flapping cursor. Maybe a "bird toggle?"

•AC•





# HOW TO START YOUR OWN AMIGA USER GROUP

BY WILLIAM SIMPSON

Los Angeles  
Amiga Users  
Group

There you sit, night after night... 14 hours on Saturday and Sunday; adjusting the color preferences on your Workbench, thinking, "Is this FUN or WHAT?". Yes, its fun playing with your Amiga. But after awhile, no matter how you use your computer, it's more fun when you can share your ideas and experiences with someone else who can APPRECIATE all the astounding things your Amiga can do.

**The Solution?** Have plenty of Amiga buddies. Ah, but there's the real problem; how do you find lots (or a least a few) people who have the same earnest belief that if a computer can't address 8 megabytes of ram it should only be discussed when the conversation turns to ancient history?

**The Answer?** Form an Amiga user group. At a minimum, you will find lots of people to appreciate your latest version of the start-up sequence; and probably someone to show you how to improve it!

Beyond socializing, opportunities to exchange information and problem solving without making a long distance call to Commodore; user groups give you quick and free access to the latest public domain software and easier access to manufacturers and distributors of Amiga hardware and software.

Especially when a computer is new on the market, like our Amiga; the need for joining or forming a user group is even greater. Right now, we are all still trying to learn how to enter the CLI; let alone learning the intricacies of our Operating System. At this moment we are creating the common knowledge of Amiga users two years hence; and rather than each of us reinventing the wheel; a user group allows us to jointly become aware of the latest methods and mutually evolve them for the benefit of us all.

There are still more benefits of user grouping including the gratification of seeing that article you wrote printed in your group newsletter, and the advantage of DISCOUNTS on purchasing that 2 megabyte memory expansion you've been dreaming about.

So, now that we've decided that it's worthwhile to form or join an Amiga user group; what is the next step? Gather together a core of people who will help.

You probably already know some other Amiga users; call them up and start making plans. If you're using a modem, get on line with folks in the vicinity. Put an ad in your local newspaper, "Amiga Users Unite!"

If all else fails try hanging around your local Amiga dealer. When you see someone browsing the Amiga software - start talking. This is a good opportunity to mention one of your greatest assets in creating a user group - your Amiga dealer. If you create a good relation with the dealer, it will

benefit both the group and dealer on a continuing basis. Start now! Get to know the manager of the store; perhaps a specific person has already been designated to work with user groups.

Don't let the fact that the dealer sells more than one kind of computer deter you. He wants to sell Amigas or he wouldn't have them in stock. If a group of Amiga users identify with his store, he will sell software, more peripherals and more computers. Very frequently, if a dealer can tell a prospective Amiga customer that there is an energetic user group in the area, the customer will feel more secure in making the purchase. Make your relationship with your dealer symbiotic. Each should benefit and enhance the other.

The first way the dealer can help is by displaying a sign up sheet for people interested in belonging to your user group. If that sheet is kept near the demonstration computer, you WILL get signatures. When you are preparing the sign up sheet, take the time to make it look good. The dealer has been given lots of pages of Amiga clip-art to use in preparing his own advertisements. Perhaps he will allow you to photocopy something appropriate as a heading for your sign up sheet.

After you get names and phone numbers of interested Amiga owners, try to determine the best time to have a meeting. Initially, you may want to meet in someone's home. This keeps the cost down and maintains an informal atmosphere for friends to gather and have some laughs.

Because of the popularity of the Amiga, you may find that even initially there are quite a few people interested in joining your group. In that case, again, talk to your Amiga dealer. He likes people to come into his store. Perhaps he has a meeting room already set aside for just such a purpose. Otherwise, he might be willing to keep his store open late on a particular night so that you can hold your meeting. If you really have a supportive dealer, he may be willing to list your meeting date and time in his advertisements in the newspaper. In our local group in Los Angeles (the Los Angeles Amiga Users Group), our supporting dealer, S.O.S. Computers, actually SUGGESTED the idea of including advertising for our group in their ads. All of the ideas in this article have been tried, and work. Give them a try!

If your membership becomes too large for the dealer to accommodate, or if the dealer is not willing to provide free space; start contacting service organizations. Banks, Savings and Loan associations, churches, the Y.M.C.A.; all of these organizations have community meeting rooms which they will allow you to use for free, or on a low rental basis. There is a room out there. Make the calls and it will appear.

After the location is determined, the next step is notifying the universe that you are having a meeting. Again, there are several methods for getting the word out. Start a phone campaign. This will work to notify the folks who signed your





## THE ORATOR SPEECH PROCESSOR for the AMIGA

Take advantage of your AMIGA's speaking powers with THE ORATOR. THE ORATOR allows you to compose text in either regular English, or using the Phoneme method (or a combination of both). A complete text editor permits you to change the spelling of words in order to get just the right sound. You have complete control over the Rate, Pitch, Tuning, Voice, and Mode of each individual phrase by simple, mouse controlled sliding bars and boxes. A phrase can be any length up to 140 characters, and at least 200 phrases can string together in a single continuous file. Your story, poem, jokes, or whatever, along with their voice settings, can be saved in a compact sequential file that you can use in your own BASIC programs. THE ORATOR also comes with THE PHONEME TUTOR, a program that makes it easy to learn the Phoneme method of text input. Includes complete documentation and a utility program for use in your own programs. Requires the AMIGA with 512K memory, one disk drive, and ABASIC or Amiga BASIC. Both versions are included on the disk.

**Price: \$39.95** postpaid C.O.D. add \$4  
(Indiana residents add 5% sales tax)  
Mail check or money order to:



THE QUALITY COTTAGE  
6301 E University Commons  
Suite 308  
South Bend, IN 46635  
(219) 234-4401

AMIGA is a trademark of Commodore-Amiga, Inc.

sign up sheet. It is certainly more effective to have a person to person conversation with everyone you can.

However, there are probably a sizeable number of Amiga owners out there that would like to attend your meeting but haven't heard about your group yet. Put an ad in your local paper. If there is a computer section in the paper, so much the better. Again, put the word out on every bulletin board (electronic and otherwise) that you can find.

I know you've read enough of this article to remember the Amiga dealer. Post notices of your first meeting. Not just in one dealer's store. Go to every Amiga dealer within 25 miles of your meeting location. And don't forget Non-Amiga dealers. If someone sells disks, or software (maybe your local bookstore), or printers; people owning Amigas will wander in from time to time.

When preparing your notice of the first meeting, be sure to mention that everyone should bring a blank disk for FREE SOFTWARE, and a checkbook for membership dues. (Please don't let them forget their checkbooks!)

So, the meeting date is set, everyone knows to bring money; are we ready? Not quite yet. Several actions should be taken to prepare for the meeting. If the first meeting goes well, everyone will be back again and again. Take the time to organize every aspect of that meeting before it begins.

First, prepare a questionnaire. It should request the Name, Address, Telephone Number and Interests of each person

attending (whether or not they intend to join the group - MORE PHONE NUMBERS!). You should also find out if the prospective member has owned other computers and the type (will old Atari users outnumber the C-64 abusers?), and a self-rating on computer sophistication - from Beginner to Programmer or Developer. With this information you will be able to create Special Interest Groups (S.I.G.s) that will accommodate the needs of everyone.

Secondly, in order to come through on your promise of FREE SOFTWARE, you've got to obtain some public domain software. If you've been reading this magazine, you know that Amazing Computing is a good source for buying Public Domain software.

Other sources include other Amiga user groups (see the listings in various computer magazines and newspapers). Most user groups are more than willing to share their library of software, usually for the price of one membership in their group. In addition, there are regular advertisements for public domain software for the Amiga in this and several other magazines.

After you've purchased some software; it must be copied. When we copied our first disk of software for our first meeting, most of us didn't have a second drive. After 40 copies, one can easily sing a few bars of the "switchin' the floppies blues". Share the load of copying software, and try to use people that have two drives! How many copies can you make? Enough to trade to everyone who shows up with a blank disk, plus an additional number for those folks who didn't know to bring a blank disk but are willing to pay for your disk or software. There's no doubt that you'll have to guess, but it's probably better to over-estimate than not to have enough disks for the people who came to the meeting to get software.

At our first meeting we had approximately 15 people attend. At our second meeting, about 35, and at our third over 60. As I said before, the Amiga is a popular computer. To put these numbers in perspective, we didn't do any advertising, other than phone calls, until the third meeting.

If you end up with extra copies of your disks, they can always be used later for people who missed your first meeting; or you can copy over the disks with the programs you intend to distribute at your next meeting.

It is especially important in the first few meetings to maintain some sense of stability. The users attending the meetings want to be sure that you are dependable and that the group is viable. To that end, try to verify that your meeting location will be available on the same day and time for at least the first 3 months. By the end of that time your group will probably have out grown the location anyway. Always be on the lookout for a larger room with better facilities.

As you will be collecting money for dues and possibly software disks, it is important to open a bank account. The best option is to try to open an account at the same bank where your personal account is placed. Frequently a bank may be unwilling to open an account for an organization until you provide by-laws and a list of officers.



If you determine that by-laws are necessary prior to opening the account, prepare tentative by-laws to hand out at the first meeting. Your prospective members won't want to take the time to read or discuss the issue of by-laws, but it can't be avoided. By explaining the necessity and having them printed up previously, there can be discussion and a vote without too much wasted time.

If you are able to promise the attendees that there will be a demonstration of some new software; they will be more willing to sit through the by-laws process. Talk to that beneficent dealer about the software demonstration, and remind him to bring along some extra copies to sell after the meeting.

Where do you get the by-laws? The best place to find them is through other user groups. In this case, it makes little difference that the other group is for Macs, PC's or C-64's. With a little modification, their by-laws will be fine as your by-laws. In future columns I will address the problem of the content of by-laws, in more detail.

As I mentioned previously, it is important to have some kind of entertainment at the first meeting. Whether software is demonstrated by the dealer, or whether a developer shows off his work in progress, or whether you schedule a hoola-hoop contest; you should have something special to conclude the evening. Also, refreshments never hurt!

After you have pinned down some special demonstration, you're ready for your first meeting. The process I've

described is not the only way to begin a user group. There are many variations on this theme. Some groups do not collect dues. Some do not have by-laws. The newsletter and even refreshments are not mandatory. Some user groups sell their public domain disk rather than trade for a blank disk. The only minimum requirement is to have semi-regular meetings.

After all this work, the lingering question in your mind may be; "Hey, is it still FUN?" The answer is yes. Each time you plan a meeting it gets easier. You will find at least a few members as dedicated as yourself, and they will be willing to help carry the responsibilities of moving the group forward. Then, just sit back and enjoy the interchange.

The best way to find volunteers is to wait for someone to comment to you "Why didn't you have a...". Your response should be, "That's a great idea, why don't you have that ready for the next meeting." If someone has the energy to complain, redirect their energy to action. If the members want something done, they will provide the manpower to do it. If they don't provide the workers they don't really want the thing done. I think this explains in part the great differences between user groups; each is a manifestation of the attitudes of its members.

Of course, if all this is too complicated, way too much trouble; you can always sit there night after night.... 14 hours on Saturday and Sunday, adjusting the color preferences.....

•AC•

#### NETCH COMPUTER PRODUCTS ANNOUNCES:

## 68881 FLOATING POINT COPROCESSOR BOARD for AMIGA

FAST - ASYNCHRONOUS 68881 OPERATION WITH A MAXIMUM CLOCK RATE OF 16.5 MHz !!  
 LOW COST DESIGN - USES INEXPENSIVE 68010 CPU EMULATING 68020 COPROCESSOR INTERFACE !  
 UPWARD SOFTWARE COMPATIBLE WITH THE 68020 - ALL SOFTWARE DEVELOPED FOR THIS BOARD  
 WILL RUN ON 68020 SYSTEMS (This is a low cost way of entering the 68020 market).  
 EXPANSION BOX NOT REQUIRED - INTERNAL PIGGYBACK CONFIGURATION REPLACES 68000 CPU.  
 SPEED INCREASE OF 300% TO 500% OR MORE FOR MATH INTENSIVE PROGRAMS !  
 COMPLETE SOFTWARE PACKAGE INCLUDED WITH ONE YEAR OF FREE UPDATES (+ handling charge).  
 OPERATION TRANSPARENT TO USERS - EMULATOR LOADED AT BOOT TIME, DOES NOT AFFECT AMIGA.

**BUNDLED SOFTWARE:** Emulator, Manx/Aztec C and Lattice C Floating Point Libraries, Assembler Tools (Allow Motorola Floating Point mnemonics to be freely mixed with 68000 code and assembled with standard AMIGA assemblers), substitute for Motorola Fast Floating Point software library.

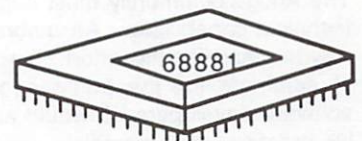
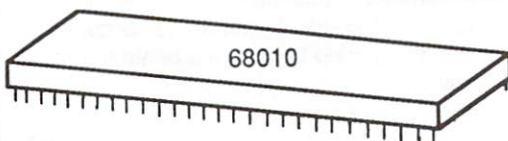
**PLANNED SUPPORT:** Fortran, Scientific Library (FFT, Matrix Algebra, etc.), and other languages.

#### INTRODUCTORY PRICING:

|       |                                                                |       |
|-------|----------------------------------------------------------------|-------|
| FPU-1 | Bare Board, Software, Documentation and Manual, parts list     | \$149 |
| FPU-2 | Assembled and Tested without 68881 Math Chip, Software, Manual | \$269 |
| FPU-3 | Assembled and Tested with 68881 Math Chip, Software, Manual    | \$459 |

Send orders to: NCP  
 PO Box 645  
 Monrovia, CA 91016  
 (818) 334-1002

Add \$3.00 for shipping & handling. Payment should be by check or Money Order. California Residents add 6% sales tax.





# The Case for a National Amiga Coalition

**By David Lipshutz**  
**Treasurer, Amuse, New York Amiga User's Group**

*David Lipshutz has his own ideas as to where an Amiga User Group should go. He expresses his belief in the need for a collective effort in supporting the Amiga. We offer this space to express his ideas and to raise our readers awareness of the support the Amiga should receive. We ask for your response and or alternative suggestions.*

The Amiga is a grassroots computer. The success of the Amiga is dependent on those of us who own and support the machine. Word of mouth is Amiga's most effective marketing tool, and the Amiga community is an invaluable ally in helping to develop new software capabilities. Amiga owners are motivated and dedicated, some might even say fanatical. Many of us have sacrificed time, energy, and money in support of the machine, often without hope of personal gain. It seems natural now to seriously consider forming a broad coalition of Amiga owners, user groups, developers, dealers, and distributors. A national organization would provide focus, structure, and resources, for the purposes of (1) increasing the Amiga's visibility/utility, (2) for encouraging the development of third-party products, and (3) education, training and dissemination of information.

Consider, for example, the press. Unfortunately, many publications have chosen to ignore the Amiga, unless they have something negative to say. This is due to an absence of information, the close-knit relationships that some publications maintain with their advertisers, and the self-perpetuating myth that the Amiga is a toy (and that Commodore is a manufacturer of toys). Despite its efforts, Commodore has not been able to fill the information vacuum and dispel these myths. A national organization could develop and nurture press contacts, provide demonstrations to reporters, distribute press releases and promotional material about new products and capabilities, and sponsor "media events". Some examples of possible media events could include: an all-Amiga conference or fair; demonstrations to institutions and professional associations; and the production and distribution to cable companies of videotapes created exclusively by the Amiga. A national organization would provide the Amiga community with a platform, and the credentials necessary to promote the Amiga effectively. In fact, the formation of a national organization could, in itself, attract media attention.

The Amiga community must begin to focus and leverage its technical capabilities. An umbrella organization is the only way to maximize this effort. It would enable the sponsorship of seminars for the purpose of attracting and educating software developers. It would allow us to assist developers by creating and providing new development tools. The

organization would be in a unique position to expand the base of public domain software by providing funds for development and an outlet for distribution. Projects could be funded, aimed at improving the operating system. In essence, a national organization would become a definitive clearing-house for technical support, documentation, bug reports and fixes, and training. We would be able to "grease the wheels of progress", so to speak.

Fred Fish and John Foust (on behalf of the Amicus Network) have both done a magnificent job of collecting, cataloging, and distributing public domain software. Their efforts have provided an enormous amount of useful and entertaining software. Imagine how much more productive they might become if they had the backing of a high-profile organization capable of supplying appreciable resources. How will it be possible to fund and provide support for a national coalition? First, through the public support of well-known and respected individuals in the Amiga community (and Commodore itself, if possible). Endorsements will provide legitimacy to the organization. Second, advertisements and articles such as this one will be sent to Amiga-dedicated and general magazines. Simultaneously, public domain disks will be commissioned and distributed that contain the same appeals for participation. Software publisher's mailing lists and dealers' lists could be the next step. In this way, a broad-based and effective drive can be mounted at a relatively modest cost. If we can induce participation by 50% of all Amiga owners at a membership fee of \$20.00, we can generate an annual budget of \$1 million or more. Other activities could increase the amount endowing a budget that would allow the organization to hire a full-time technical and marketing staff that would help carry out the organization's objectives.

Ours is a mutually interdependent community, each interest group generates strength from drawing on the resources of the others. Currently, the most important thing we can do is to strengthen our own ties, and reach out en masse to attract the entire computer establishment. A national nonprofit organization is the ideal means to accomplish these ends, and in so doing, help to secure a bright future for the Amiga and ourselves.

•AC•



# Amiga Users Groups

North American Digital BBS  
3512 East Grant Road  
Tucson, AZ 85716  
Voice (602) 323-7897  
Please don't call during business hours

1st Amiga Users Group  
543 Old Colony Road  
San Carlos, CA 94070  
(415) 595-5452  
Robo City News \$12.00/Year  
AmigaBoard! BBS  
Stuart Sanders, Malibu, CA  
Voice (213) 477-4728

Gene's Machine BBS  
Ridgecrest, CA

Los Angeles Amiga User's Group  
Mr. William Simpson  
1711 Altivo Way  
Los Angeles, CA 90026  
(213) 661-7959  
Baha'i Faith Center  
5755 Rodeo Rd, Los Angeles  
3rd Tuesday of every month, @7:30 pm

Sacramento Area Computer Club  
2nd Tuesday, Fair Oaks Library  
1-916-944-7400 George Leone

San Fernando Valley  
Amiga User's Group  
Northridge, CA  
(818) 786-8624  
Valley Federal Savings Bank,  
corner of Reseda & Nordhof,  
Northridge. Every first Tuesday of the  
month, @6:30pm

Valley Computer Club  
Amberse M. Banks / Librarian  
505 Ryan Avenue  
Modesto, CA 95350

National Capital Area User Group  
P.O. Box 18088  
Washington, DC 20036-8088  
GURU \$18.00/Year  
George Washington University, Bldg. C  
3rd Saturday of every month,  
@1:00pm

Amiga Info Xchange BBS  
Charles R. Goodman  
225 S. Clair Drive  
Panama City, FL 32401  
Voice (904) 763-4926

Central Florida Amiga Club  
Mr. T. Lee Kidwell / Founder  
1056 Neely Street  
Oviedo, FL 32765  
CFAC Newsletter \$25.00/Year  
Additional Family Members, \$7.50 ea.  
Micros Etc.  
515 E. Hwy. 436  
General Meetings on 1st & 3rd  
Tuesday of each month

Space Coast AMIGA User's Group  
Mike Dalton / President  
P.O. Box 2098  
Merritt Island, FL 32952  
(305) 452-4737  
Space Coast AUG Newsletter  
\$15/Year, \$10/Half Year, \$2/ Month  
Merritt Island Public Library  
Every 2nd Wednesday of the month @  
6:30pm

Tampa Bay Amiga Group  
Billy W. Combs / Treasurer  
920 South Rome Avenue  
Tampa, FL 33606  
(813) 251-2480

Amiga Atlanta  
Andre Frech  
Box 7724 Atlanta, GA 30357  
(404) 676-0384  
The Workbench \$30.00/Year

Eagle Rock Commodore Computer Club  
James R. White, Amiga Librarian  
2100 Belmont Avenue  
Idaho Falls, ID (208) 524-5464  
Eagle Rock News  
Idaho Falls Library  
Third Friday of every month @ 7:30pm

AAUG Algonam Amiga Users  
CRS in Algonam II  
2nd Monday @ 7:00 PM  
1 312 658 3040 1 312 658 6136

Commodore Hardware User's Group  
Greg Chancey  
1322 Fairview Drive  
Greenfield, IN 46140  
(317) 462-3748  
The C\*H\*U\*G \$15.00/Year  
Warren High School Cafeteria,  
9500 E. 16th Street, Indianapolis  
Third Wednesday of each month,  
@6:00 pm

Amiga SIG / Public Domain Library  
Fort Wayne Amiga Users Group  
Tony Vassiliadis / President  
183 Oak Park Drive  
Roanoke, IN 46783  
(219) 672-3001  
IPFW Campus, Student Union Building,  
Rm. 226  
Date Usually Changes / @6:30pm

Indiana Amiga Advisors  
Dennis Graham / President  
912 South Brown Avenue  
Terre Haute, IN 47803  
(812) 234-5099  
Indiana Amiga Advisors Newsletter  
\$12.00/Year  
Spectra Computer Systems  
1100 Ohio Street  
Terre Haute  
4th Tuesday of every month, @7:30pm

Amiga group of  
the Boston Computer Society  
Dept. of Transportation, Kendle Sq.  
Amiga users meet 3rd Tuesday  
Amiga Tech Group meets 1st Tuesday  
BBS Window 617 868 1430  
BBS Wonderland 617 665 3796

Baltimore Amiga Users/Developers  
(BAUD)  
Ed Hopper / Club Coordinator  
243 West 31st Street  
Baltimore, MD 21211  
(301) 467-1034  
Commodore User's Medium

Baltimore Area Computer Club (CUM-  
BACC)  
William J. Kolodner / President  
P.O. Box 479  
Reisterstown, MD 21136-9998  
Compuserve 75026,1074

Amiga SIG / PDS Library  
...AAAmiga Eoin Cain  
P.O. Box 4272  
Ann Arbor, MI  
(313) 662-1520

Amiga User Group  
Pontiac Michigan  
Edmunton Computer Center, Inc.  
2548 Orchard Lake Rd.  
Pontiac, MI 48053-2435  
Call for time  
1 313 681 8800  
Les Jenkins



Gateway Amiga Club, Inc.  
14850 Phelps  
Bridgeton, MO 63044  
(314) 739-5181  
Bridgeton Trails Library  
3455 McKelvey Rd  
Every first Wednesday of each  
month @ 7:20pm

The Amigans  
P.O. Box 411  
Hatteras, NC 27943  
Apprentice & Journeyman  
\$25.00/Year

Greater Omaha Commodore  
User's Group  
Fred Layberger / President  
P.O. Box 241155  
Omaha, NE 68124  
(402) 896-0136  
Amiga SIG

Nebraska Amiga Users Group  
Paul Loto  
P.O. Box 2414  
Lincoln, NE 68502

Jersey Amiga Users Group  
Perry Kivolowitz 271-4522  
Hill Center NJ  
Eric Lavitsky 745-2839

New Mexico Commodore User's Group  
Walt Stanley  
P.O. Box 37127  
Albuquerque, NM 87176  
Dimensions \$12.00/Year

AMICUS?H.V., Hudson Valley  
P.O. Box 1168  
Wappingers Falls, NY 12590  
Jim Meyer 914 297 8759  
Ralph Blanchette 914 266 5606  
Computer Emporium  
Main St., New Platz NY  
last Thursday of the Month, &PM  
\$10 per year  
Newsletter, "Vox Amica"

Amiga Mouse User's Group BBS  
Joseph L. Rothman  
P.O. Box 148  
Central Islip, NY 11722

AMuse, New York Amiga Users Group  
Joe Lowery  
151 1st Avenue, Suite 182  
New York, NY 10003  
(212) 269-4879 5am-3am 7 days

P.S. 122, 150 1st Ave. at 9th St., NYC  
1st and 3rd Thursday of each month at  
8PM  
1ST Thurs. for beginners and end-  
users  
2ND Thurs. for developers and  
hackers

Computer Users Group of Rochester  
Steve Collins / President  
P.O. Box 23463  
Rochester, NY 14692  
(716) 334-8991

Economy Amiga User's Group  
189 Pinewood Road  
Hartsdale, NY 10530  
(914) 683-8912  
EAUG Newsletter

ATOP Development BBS  
11914 Girdled  
Concord, OH 44077  
Voice (216) 352-8471

Ohio Valley Amiga User's Group  
Eric Hanson / President  
3115 Bellewood Avenue  
Cincinnati, OH 45213-1603  
(513) 351-0945  
Location Varies  
Last Tuesday of each month @7:00  
pm

Amiga Computer Enthusiasts  
Mark Fulton / President  
1111 N. St. Charles #16  
Oklahoma City, OK 73127  
(405) 787-7273  
Oklahoma City Community College  
7777 South May Avenue (Conference  
Room A)  
First Tuesday of every month @  
6:00pm

The Northwest Amiga Group  
P.O. Box 1140  
Oregon City, OR 97045  
NAG RAG \$2.00 per month or  
\$24.00 per year  
Bonneville Power Administration  
Auditorium  
3rd Tuesday of every Month @ 7:00pm  
All memberships are due on January  
1st

East Hills Amiga Group  
Pittsburg, PA  
Data Softique Computerware  
928 Presque Isle Drive  
1st Wednesday of every month,  
@7:00 pm

Carolina Amiga Users (CAUSERS)  
Ray Marsh / President  
2224 Airport Road  
West Columbia, SC 29169  
(803) 782-2213  
GADGET \$5.00 initial fee  
Avcom International, Beacon Hill Block  
2224 Airport Road, West Columbia, SC  
2nd & 4th Wednesday of every month,  
@7:30pm

Amiga User's Club of Houston  
Joseph Manby  
6525 S. Gessner #3100  
Houston, TX 77036  
(713) 981-1507  
Computer Age  
7535 Westheimer, Houston  
2nd & 4th Sunday of the month,  
from 2:00pm-5:00pm

Olympia Amiga Users Group  
Noah Sherman  
4104 Amber Ct. SE  
Olympia, WA 98501  
(206) 943-8257  
South Sound Community Center  
Once a month

Washington Amiga User Group "WAG"  
Pres. Richard Medved  
2nd & 4th Thursday 7:30  
1-206-659-9044

Amiga BitMappers  
Mr. Clinton Kurek  
P.O. Box 1641  
Milwaukee, WI 53201

Public Service Announcement - BBS  
Dorothy Dean  
Milwaukee, WI

Amiga SIG  
London Amiga Club  
Bob Rutter  
London, Ontario, Canada  
451-0228  
Althouse College, Room 2029, London  
Ontario  
Second Monday of each month, @7:00  
pm

Pacific Northwest Amiga Association  
10851 Shellbridge Way  
Richmond, B.C., Canada V6X 2W8  
(604) 273-2243  
PaNorAma Newsletter \$25.00 per year  
General Meeting - 2nd Wednesday of  
each month, @7:30 pm

•AC•



# AmigaBASIC Tutorial

By Kelly Kaufman  
CIS [70206,640]

In this installment, we'll go a bit deeper into variables and some of the different numerical functions you can perform on them. The next 6 functions are the "easy" stuff, following these 6, we'll go into the trigonometric functions of MBasic. And first up is....

**ABS:** The "ABS" function returns the absolute value of an argument given in parenthesis. Example:

X=ABS(-38)

In this example, X would equal 38. Note that this feature DOES retain the decimal point and any places behind it.

**CINT:** CINT rounds off an expression to the nearest integer. Example:

X=CINT(12.8323)

X now would equal 13...rounding 12.8323 off.

**INT:** INT drops off the decimal point and keeps the integer value. It DOES NOT round numbers off. For rounding, use CINT. Example:

X=INT(12.8323) X is now 12, dropping the .8323.

**RND:** You'll probably be using RND a lot if you are going to be doing any game programming in MBasic. This function will generate a random number between 0 and 1. Here are a few examples:

X=INT(RND\*100) Generates a random number between 0 and 100.

X=RND\*100 Generates a random number between 0 and 100, but will not be an integer. It is advisable before your program does it's random number generation, to include the statement "RANDOMIZE TIMER." This will "SEED" the random number generator, thereby giving you more authentic random numbers.

**SGN:** SGN returns the value of a number as it relates to 0. For instance, if an argument is less than 0, a -1 will be returned. If the argument is equal to 0 then a 0 will be returned. And if the value of the argument is greater than 0, a 1 will be returned. Examples:

X=SGN(3) X will be 1  
X=SGN(-3) X will be -1  
X=SGN(0) X will be 0

**SQR:** SQR returns the square root of a number. Example:

X=SQR(9) X will be 3, the square root of 9.

Well so much for the easy stuff. Now here's the list of the trigonometric functions of MBasic, just like I promised.

**ATN:** ATN returns the arc tangent of the argument. Example:

X=ATN(2) X will be 1.107149---the arc tangent of 2.

**COS:** COS returns the cosine of an argument in radians. Example:

X=COS(100) X will be .8623189---the cosine of 100.

**EXP:** EXP returns the base of natural logarithms to the power of the argument. Example:

X=EXP(3) X will be 20.08554--The natural base.

**LOG:** LOG returns the logarithm of the argument. Example:  
X=LOG(1.6) X will be .4700037, the logarithm of 1.6.

**SIN:** The SIN function will return the sine of the following argument. Example:

X=SIN(100) X will be -.5063657, the sine of 100.

**TAN:** TAN returns the tangent of the argument. Example:  
X=TAN(1.777) X will now equal -4.780646, the tangent of 1.777.

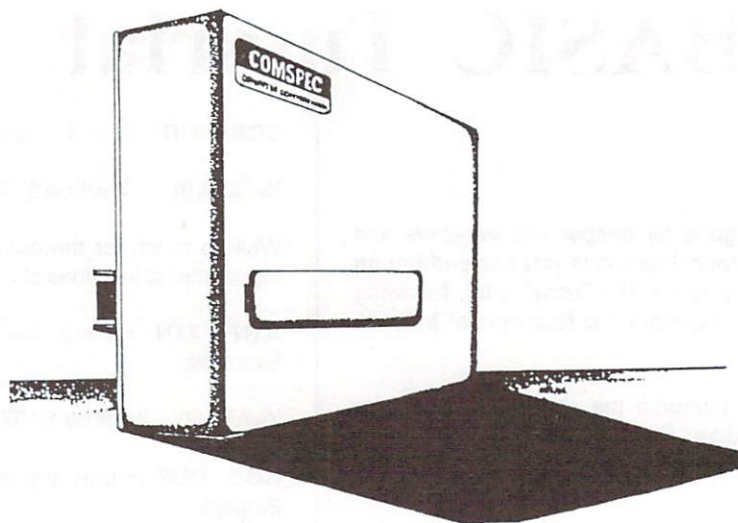
If your trigonometric needs exceed the commands given in MBasic, there is a table on page A-17 of your MBasic manual that describes formulas for 21 other trigonometric functions. Another function that you will be using as you start making larger programs, is the FRE function. This function informs you as to how much memory is left in your Amiga. There are three areas we can look into. The entire system, the Stack, and Basic programming area. We won't concern ourselves with the Stack for now...that will be in a later installment of this tutorial. Here is the method for checking your remaining memory.

X=FRE(-1) Returns the total number of bytes left in the ENTIRE system.

X=FRE(1) Returns the number of bytes left in MBasic's List window. That is, the total number of characters left to program in. For example if it returned 5000, then you would have space left for 5000 more characters worth of programming.

These functions should help you in your number crunching abilities on your Amiga. So what are you waiting for??? Go ahead and turn your Amiga on and figure out the meaning of the universe!!!





## 2 Megabyte Ram Board For The Amiga PC

Comspec Communications Inc. is proud to announce an exciting new product. The MICROSHARE AX 2000 - a 2 megabyte ram board for the Commodore Amiga PC.

The ability of the AX 2000 to incorporate 2 megabytes of ram and allow the user growth potential by the cascading of additional AX 2000s, now gives the Amiga PC vastly extended computing power.

Mounting on the right hand side of the Amiga PC the AX 2000 connects to the expansion port. Additional peripherals which would normally be connected to the expansion port can then be connected on the expansion port of the AX 2000. Standard expansion bus architecture was used in the design of the AX 2000, thereby insuring compatibility with all peripherals.

The AX 2000 can be used to increase ram memory, or as a fast ram drive. When used to increase ram memory the AX 2000 will greatly enhance the multi-tasking feature of the Amiga PC. With more memory available to the system it is possible to have more tasks running simultaneously or a greater amount of memory available to each of the tasks. Software developers will find the AX 2000 a must, allowing a great reduction in compile/assembly time, thus leading to a real savings in terms of development costs.

The MICROSHARE AX 2000 is available from stock, for immediate delivery.

|                         |          |           |                             |
|-------------------------|----------|-----------|-----------------------------|
| Suggested Retail Price: | Canadian | \$1276.00 | Cdn Funds                   |
|                         | U.S.     | \$899.00  | U.S. Funds (F.O.B. Toronto) |

Dealer Enquiries Invited

For further information contact:

Comspec Communications Inc.  
153 Bridgeland Ave., Unit 5  
Toronto, Ontario M6A 2Y6  
(416)-787-0617

Amiga is a registered trademark of Commodore Business Machines



# Mailing List

## In AmigaBASIC

By Kelly Kaufman

One of the best uses of a computer is to track information. This mailing list program will hold a list of peoples names, addresses, phone numbers (both work and home), city, state and zip codes. After information has been keyed in, it can then be printed out on mailing labels for use on letters, packages, or whatever use you may have. This program requires AmigaBASIC. For single drive users, this program will work best with AmigaBASIC and this program on your workbench diskette. Type the program listing into the LIST window and SAVE IT BEFORE YOU RUN IT!!! If any section of the program doesn't perform properly, compare your listing to the one listed below. Good luck!

### Mailing List Prog:

```
' Mailing List by Kelly Kauffman
' May 31, 1986
'
WINDOW 1,"Mailing List V1.0",,16
PRINT
PRINT "Enter the Filename of your Mailing
List"
PRINT "or Press [RETURN] to create a new
List."
PRINT
PRINT
INPUT fil$
IF fil$="" THEN GOTO mnu
OPEN fil$ FOR INPUT AS #1
INPUT #1,maxrec
dimsize=maxrec+100 'allows for 100 more
records to be entered
' in one session than was
originally in the 'file. Change as needed.
A lower number 'will not eat up as much
memory.
DIM
F$(dimsize),ln$(dimsize),a$(dimsize),c$(dim
size)
DIM
s$(dimsize),z$(dimsize),hp$(dimsize),wp$(dim
size)
CLS
LOCATE 15,30
PRINT "Total Records:";maxrec
FOR rec=1 TO maxrec
LOCATE 17,29
PRINT "Reading Record:";rec
LINE INPUT #1,ln$(rec)
LINE INPUT #1,F$(rec)
LINE INPUT #1,a$(rec)
LINE INPUT #1,c$(rec)
LINE INPUT #1,s$(rec)
LINE INPUT #1,z$(rec)
LINE INPUT #1,hp$(rec)
LINE INPUT #1,wp$(rec)
NEXT rec
rec=maxrec
CLOSE #1

mnu:
CLS
COLOR 0,1
LOCATE 1,34
PRINT " " '11 spaces
LOCATE 2,34
PRINT " ";
COLOR 1,0
PRINT "Mail List";
COLOR 0,1
PRINT " "
LOCATE 3,34
PRINT " " '11 spaces
COLOR 1,0
```

```
PRINT SPC(14);"Choose an option by
pressing the appropriate key:"
PRINT
PRINT "E)dit a record."
PRINT
PRINT SPC(4);"D)isplay a record."
PRINT
PRINT SPC(8);"P)rint a record."
PRINT
PRINT SPC(12);"M)ailing label print."
PRINT
PRINT SPC(16);"L)eave Mail List."
PRINT
PRINT SPC(20);"S)tore records to disk."
PRINT
PRINT SPC(24);"+"Add a record."
PRINT
PRINT SPC(28);"-"Delete a record."
PRINT
PRINT "Press a key (E D P M L S A + -)";
```

```
menuchoose:
r$=INKEY$
IF r$="" THEN GOTO menuchoose
r$=UCASE$(r$)
IF r$="+" THEN GOTO add
IF r$="M" THEN GOTO mail
IF r$="S" THEN GOTO store
IF r$="L" THEN GOTO leave
IF r$="E" THEN GOTO ed
IF r$="D" THEN GOTO display
IF r$="P" THEN GOTO prt
IF r$="-" THEN GOTO del
'only gets through here if they didn't
press
'one of the options.
BEEP
GOTO menuchoose
```

```
add:
GOSUB displayform
rec=rec+1
LOCATE 4,14
LINE INPUT " ",F$(rec)
LOCATE 6,14
LINE INPUT " ",ln$(rec)
LOCATE 8,14
LINE INPUT " ",a$(rec)
LOCATE 10,14
LINE INPUT " ",c$(rec)
LOCATE 12,14
LINE INPUT " ",s$(rec)
LOCATE 14,14
LINE INPUT " ",z$(rec)
LOCATE 16,14
LINE INPUT " ",hp$(rec)
LOCATE 18,14
LINE INPUT " ",wp$(rec)
LOCATE 21,1
mesq$="Is everything Okay"
GOSUB yn
IF yn$="N" THEN
rec=rec-1
GOTO add
END IF
mesq$="Add Another Record"
GOSUB yn
IF yn$="N" THEN GOTO mnu
GOTO add
```

```
displayform:
CLS
COLOR 0,1
PRINT
PRINT
PRINT " First Name: "
PRINT " Last Name: "
PRINT " Address: "
PRINT " City: "
PRINT
PRINT " State: "
PRINT " Zip: "
PRINT " Home Phone: "
PRINT
```



# KILL A MOUSE GO TO JAIL!

MOUSE WASH IS A SPECIALLY  
DESIGNED BALL WHICH:

- CLEANS THE INSIDES OF YOUR MOUSE!
- SAVES YOU TIME AND MONEY!
- CAN BE USED HUNDREDS OF TIMES!
- NEEDS NO CHEMICALS

SO TREAT YOUR MOUSE  
TO MOUSE WASH TODAY

LET SAYIT! READ YOUR TEXT FILES TO YOU!  
IT READS MOST STANDARD TEXT FILES ALoud!

MOUSE WASH \$7.95 -- SAYIT \$14.95  
\$2.00 SHIPPING \$3.00 FOR C.O.D.

T&L PRODUCTS, 2645 Wilson Street  
Carlsbad, CA 92008 (619) 729-4020

```
PRINT " Work Phone: "
PRINT
COLOR 1,0
RETURN

yn:
msg$=msg$+" (Y/N/X)?"
LOCATE 22,1
PRINT "

"; '50 spaces
LOCATE 22,1
PRINT msg$;
1 yn$=INKEY$
IF yn$="" THEN GOTO 1
yn$=UCASE$(yn$)
IF yn$="X" THEN GOTO mnu
IF yn$<>"Y" AND yn$<>"N" THEN GOTO 1
RETURN

displayrec:
LOCATE 4,15
PRINT F$(dsp)
LOCATE 6,15
PRINT ln$(dsp)
LOCATE 8,15
PRINT a$(dsp)
LOCATE 10,15
PRINT c$(dsp)
LOCATE 12,15
PRINT s$(dsp)
LOCATE 14,15
PRINT z$(dsp)
LOCATE 16,15
PRINT hp$(dsp)
LOCATE 18,15
PRINT wp$(dsp)
RETURN

display:
CLS
PRINT "Search by F)first name or L)ast
name or A)ll records"
2 x$=INKEY$
IF x$="" THEN GOTO 2
x$=UCASE$(x$)
IF x$<>"F" AND x$<>"L" AND x$<>"A" THEN
GOTO display
IF x$<>"A" THEN
```

```
CLS
PRINT "What name do you want to
display?"
PRINT
PRINT "Choose name by entering a search
pattern for the last name. If"
PRINT "I find a match within the name I
will display it for you."
PRINT
PRINT
INPUT "Search Key = ",srch$
srch$=UCASE$(srch$)
END IF
FOR i=1 TO rec
IF x$="A" THEN place=1
IF x$="L" THEN
place=INSTR(UCASE$(ln$(i)),srch$)
IF x$="F" THEN
place=INSTR(UCASE$(F$(i)),srch$)
IF place>0 THEN
dsp=i
IF ln$(i)="DELETED" THEN 9
GOSUB displayform
GOSUB displayrec
msg$="Okay to Continue"
GOSUB yn
IF yn$="N" THEN GOTO mnu
END IF
9 NEXT i
CLS
LOCATE 15,35
COLOR 0,1
PRINT " No more matches. "
COLOR 1,0
FOR rest=1 TO 3000
NEXT rest
GOTO mnu

ed:
CLS
PRINT "Do you want to search for the name
to edit on the F)first or"
PRINT "L)ast name field?"
3 x$=INKEY$
IF x$="" THEN 3
x$=UCASE$(x$)
IF x$<>"F" AND x$<>"L" THEN 3
PRINT
PRINT "What name do you want to edit?"
PRINT
PRINT "Choose a set of letters which will
be searched on in the name"
PRINT "field. If a match is found, I will
present it to you and you can"
PRINT "decide if it is the record you want
to edit."
PRINT
PRINT
INPUT "What letters to search on?"
",srch$
srch$=UCASE$(srch$)
FOR i=1 TO rec
IF x$="F" THEN
x=INSTR(UCASE$(F$(i)),srch$)
IF x$="L" THEN
x=INSTR(UCASE$(ln$(i)),srch$)
IF x>0 THEN
GOSUB displayform
dsp=i
GOSUB displayrec
msg$="Edit THIS record"
GOSUB yn
IF yn$="N" THEN
msg$="Continue Search"
GOSUB yn
IF yn$="N" THEN GOTO mnu
IF yn$="Y" THEN GOTO 4
ELSE
GOTO ed
END IF
END IF
4 NEXT i

ed:
CLS
PRINT "When editing, the cursor will
appear just under the line that"
PRINT "is ready to be edited. If you wish
to keep the line the same,"
PRINT "don't type it again, just press
return and I'll keep it the same."
PRINT
PRINT "Press a key to continue."
5 x$=INKEY$
IF x$="" THEN GOTO 5
GOSUB displayform
GOSUB displayrec
```



```

LOCATE 5,14
LINE INPUT " ",chg$
IF chg$<>" THEN F$(dsp)=chg$
LOCATE 7,14
LINE INPUT " ",chg$
IF chg$<>" THEN ln$(dsp)=chg$
LOCATE 9,14
LINE INPUT " ",chg$
IF chg$<>" THEN a$(dsp)=chg$
LOCATE 11,14
LINE INPUT " ",chg$
IF chg$<>" THEN c$(dsp)=chg$
LOCATE 13,14
LINE INPUT " ",chg$
IF chg$<>" THEN s$(dsp)=chg$
LOCATE 15,14
LINE INPUT " ",chg$
IF chg$<>" THEN z$(dsp)=chg$
LOCATE 17,14
LINE INPUT " ",chg$
IF chg$<>" THEN hp$(dsp)=chg$
LOCATE 19,14
LINE INPUT " ",chg$
IF chg$<>" THEN wp$(dsp)=chg$
GOSUB displayform
GOSUB displayrec
msg$="All edits correct"
GOSUB yn
IF yn$="Y" THEN mnu
IF yn$="N" THEN edt

mail:
CLS
PRINT "Which records do you want to
print?"
PRINT
PRINT "Do you want to:"
PRINT
PRINT "1. Print all records."
PRINT
PRINT "2. Print some based on a search in
the last name field."
PRINT
PRINT "3. Print some based on a search in
the first name field."
PRINT
PRINT "4. Go through all records and
choose each one to be printed."
PRINT
PRINT
INPUT "Choice: ",chc
IF chc>4 OR chc<1 THEN GOTO mnu

prtsome:
CLS
PRINT
INPUT "How many lines between labels?"
spaces=
PRINT
PRINT
IF chc=2 THEN INPUT "What letters to
search on in the last name field? ",srch$
IF chc=3 THEN INPUT "What letters to
search on in the first name field?
",srch$
srch$=UCASE$(srch$)
FOR i=1 TO rec
IF chc=2 THEN
place=INSTR(UCASE$(ln$(i)),srch$)
IF chc=3 THEN
place=INSTR(UCASE$(F$(i)),srch$)
IF chc=4 OR chc=1 THEN place=1
IF place>0 THEN
IF chc=4 THEN
IF ln$(i)="DELETED" THEN GOTO 6
GOSUB displayform
dsp=i
GOSUB displayrec
msg$="Print this one"
GOSUB yn
IF yn$="N" THEN 6
END IF
LPRINT F$(i);
IF F$(i)<>" THEN LPRINT " ";
LPRINT ln$(i)
LPRINT a$(i)
LPRINT c$(i);", ";s$(i);" ";z$(i)
FOR g=1 TO spaces
LPRINT
NEXT g
END IF
6 NEXT i
GOTO mnu

prt:
CLS
PRINT
PRINT "Print what record? (Note: Search

```

## ADFO

### AMIGA DISK FILE ORGANIZER

Having trouble finding that file somewhere in your stack of floppys? Can't find all the copies of a particular file?

ADFO maintains a database of the directories disknames and filenames from your collection of disks. Fast response inquiries return location and last update information. Printer interface. Uses CLI or Workbench.

512K ram and 2 drives recommended—\$59.95

Include \$3.50 S & H  
Mastercard/Visa Accepted  
Sorry, No COD

Calif. Residents Add 6½% Sales Tax

*Westcom Industries*

3386 Floyd  
Los Angeles, CA 90068  
(213) 851-4868

Order phone 1 800 621-0849 Ext. 494

```

based on last name.);
INPUT srch$
srch$=UCASE$(srch$)
FOR i=1 TO rec
x=INSTR(UCASE$(ln$(i)),srch$)
IF x>0 THEN
GOSUB displayform
dsp=i
GOSUB displayrec
msg$="Print THIS record"
GOSUB yn
IF yn$="N" THEN CLS:GOTO 7
LPRINT
LPRINT " First Name: ";F$(i)
LPRINT
LPRINT " Last Name: ";ln$(i)
LPRINT
LPRINT " Address: ";a$(i)
LPRINT
LPRINT " City: ";c$(i)
LPRINT
LPRINT " State: ";s$(i)
LPRINT
LPRINT " Zip: ";z$(i)
LPRINT
LPRINT " Home Phone: ";hp$(i)
LPRINT
LPRINT " Work Phone: ";wp$(i)
LPRINT
END IF
7 NEXT i
CLS
LOCATE 15,35
COLOR 0,1
PRINT " No more matches. "
COLOR 1,0
FOR rest=1 TO 3000
NEXT rest
GOTO mnu

del:
CLS
PRINT "Delete what name? (Note: Search
based on last name.);
INPUT srch$
srch$=UCASE$(srch$)
FOR i=1 TO rec
x=INSTR(UCASE$(ln$(i)),srch$)
IF x>0 THEN

```



```

GOSUB displayform
dsp=i
GOSUB displayrec
msg$="Delete THIS record"
GOSUB yn
IF yn$="N" THEN CLS:GOTO 8
IF yn$="Y" THEN
  ln$(i)="DELETED"
  FS$(i)="DELETED"
END IF
8 NEXT i
CLS
LOCATE 15,35
COLOR 0,1
PRINT "No more matches. "
COLOR 1,0
FOR rest=1 TO 3000
NEXT rest
GOTO mnu

store:
CLS
msg$="Ready to store this mailing list"
GOSUB yn
IF yn$="N" THEN GOTO mnu
CLS
PRINT
PRINT "What Filename ";
INPUT F$
IF UCASE$(F$)=UCASE$(fil$) THEN
  PRINT
  msg$="Save over the old file"
  GOSUB yn
  IF yn$="N" THEN GOTO mnu
END IF
CLS
PRINT "Opening File..."
CLOSE 1
OPEN F$ FOR OUTPUT AS #1
PRINT
PRINT "Writing Total # of Records..."
FOR i=1 TO recs
  IF ln$(i) <> "DELETED" THEN recs=recs+1
NEXT i
PRINT "There are";recs;"to save."

```

```

PRINT #1,recs
recs=0
FOR i=1 TO recs
  IF ln$(i)="DELETED" THEN 10
  recs=recs+1
  LOCATE 10,15
  PRINT "Saving record #";recs
  PRINT #1,ln$(i)
  PRINT #1,FS$(i)
  PRINT #1,a$(i)
  PRINT #1,c$(i)
  PRINT #1,s$(i)
  PRINT #1,z$(i)
  PRINT #1,hps$(i)
  PRINT #1,wp$(i)
10 NEXT i
PRINT
PRINT "Closing File up..."
CLOSE #1
PRINT
PRINT "Done."
FOR rest=1 TO 3000
NEXT rest
GOTO mnu

leave:
CLS
msg$="Are you sure you want to leave
now"
GOSUB yn
IF yn$="N" THEN GOTO mnu
CLS
PRINT
PRINT "Do you want to Store this mailing
list off before leaving"
GOSUB yn
IF yn$="N" THEN SYSTEM
GOTO store

```

•AC•

# Conversation With A Computer

(It's a lot of fun, a brain teaser and a programming guide too!)

*"Very highly recommended by me is Conversation With A Computer, from Jenday Software, a set of games and conversation written in Amiga Basic, and shipped with the source code provided. It is entertaining, amusing, thought provoking, and just plain fun. If you have any interest in programming in BASIC on the Amiga, this is a must have for the examples."*

—MATTHEW LEEDS, Commodore Microcomputers

This program really shows off Amiga's talents: lots of color graphics, mouse routines, voice synthesis, sound and animation. The 2,000 lines of Amiga Basic can be listed to screen or printer. The documentation describes in detail, module by module, how it all works. There is a coded example of virtually every one of Amiga Basic's powerful features.

You'll be challenged to three mind games. Memory Test will drive you to drink. Battle of Numbers and Pegboard are two of the most elegant logic games of all time. It's your brain against Amiga's silicon!

The program is professionally packaged with comprehensive typeset documentation. It requires 512K. It is not copy protected. Now includes an introduction to the C language.

**JENDAY**  
SOFTWARE

P.O. Box 4313  
Garden Grove, CA 92642

All orders are shipped by first class  
mail within 24 hours of receiving  
your personal check or money  
order.

DEALER INQUIRIES INVITED

(714) 636-3378

**ORDER FORM**  
Please rush me Conversation With A Computer with source code.  
Enclosed please find a check for \$29.50 plus \$2.50 postage and  
handling. (Californians add \$1.77 sales tax.)  
Name \_\_\_\_\_  
Address \_\_\_\_\_  
City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_



# Pointer Image Editor

By **Stephen R. Pietrowicz**

CIS: 73047,2313

People Link: DR RITZ

Usenet: ...lihn4!pur-ee!gould!houligan!srp

Many programs for the Amiga change the default pointer sprite from the familiar arrow to another image while you use that program. Changing your default pointer is a relatively simple task in C, but what if you are not familiar with C or don't own a C compiler?

By including some of the Amiga libraries into your AmigaBasic programs, you can access the subroutines to create your own mouse pointer images! The same system calls that C uses to change pointer images can be accessed through AmigaBasic. Including the exec and intuition libraries in your program with the LIBRARY statement allows you to access the necessary routines to change your pointer.

Creating a pointer is simple. First, you have to allocate enough memory for your pointer using the AllocMem() function. Then, you have to load that memory with the sprite image of the pointer you want to create. After loading the image data, call the intuition SetPointer() function, and voila! Your own pointer!

Creating the pointer image data can be a bit difficult to do. The AmigaBasic program listed at the end of this article takes away the drudgery of creating that data. It allows you to change the default colors used by your default pointer sprite, create pointer sprites that are up to 16 pixels wide and 156 pixels long, and load/save the pointer images you create.

There are several areas on the screen that you will use to design your pointer image. The large box to the left side of the screen is used as your palette. The four boxes below are the colors you can use to draw with. The box at the right side of the screen is labeled "RGB Settings". After selecting the color you want to change, adjust the RGB settings to the color you want. (It works similarly to drawing programs that are available). The gadget in the middle of the screen lets you scroll to the part of the image you want to work with. There are also two menu lists with 7 functions to use.

The first menu list has options to load a pointer image from a file, save the current pointer image to a file, clear the current pointer palette, and quit the program.

When loading or saving a pointer image, a requestor box will appear to type the file's name. Type any alphanumeric name

in the requestor box. (Hit the ESC key if you want to exit this mode). The pointer image is stored in the following format: the first three lines contain the RGB values used to create the 17th, 18th, and 19th hardware color register entries. Simply load these values by using the AmigaBasic PALETTE statement. The next statement holds the height of the pointer image.

The next entries in the file are with values used to make up the image. There are two values/line, and one line for every pixel in height. These are the values that are stored into memory. The last line contains the program's representation of the X and Y offsets of the "hot spot" of the pointer. (The "hot spot" is the point within the pointer image used to determine where the pointer is when a mouse key is pressed, and is denoted as a yellow square on the pointer palette).

The second menu list has options to test the pointer image you've created, reset the test pointer to the default program pointer, and reset the hot spot to a different part of the sprite. A word of warning when setting the "hot spot". Because of the way that the Amiga handles the mouse pointer, setting the "hot spot" near the bottom of the sprite image will make it disappear when the pointer moves near the top of the screen. It's generally a good idea to only set the "hot spot" within the first 15 pixels from the top of the image.

The two system calls that deal with the pointer are SetPointer and ClearPointer. The format of the SetPointer function is:

```
CALL SetPointer  
      (window, image, height, width, HotX, HotY)
```

where 'window' is the address of a window structure. You get that address by the WINDOW(7) function from AmigaBasic. WINDOW(7) returns a pointer to the intuition window. 'image' is the address of the mouse pointer image. The address you pass to SetPointer is the address of the first byte of the memory return from the AllocMem function. 'height' is the height of the mouse pointer image. 'width' is the width of the mouse pointer image. 'HotX' is the X-Offset of your mouse pointer image from the "hot spot". 'HotY' is the Y-Offset of your mouse pointer image from the "hot spot".

The "hot spot" offsets are normally negative values. The upper-left hand corner of the sprite has HotX and HotY values of 0,0. The upper-right hand corner of the sprite has values -15,0. A "hot spot" placed 7 pixels over and 9 pixels down has a value of -7,-9. For an example of how to use SetPointer, examine the DefaultPointer section in the program listing.



The ClearPointer system call will return the pointer that you've changed on the screen back to the original system default mouse pointer image. The call in AmigaBasic is simple:

```
CALL ClearPointer(window)
```

where 'window' is the same address of the window structure as used in the SetPointer system call. Here are some ideas for you to try in your own programs: By keeping track of which direction the mouse is moving, you can change the pointer image to point in that direction. You don't have to be limited to arrows. For example, you can have a pointer that looks like a pointing finger, one image for each direction. You can create animated pointers in your own programs by continually changing images. You can create a flying bird, a bouncing ball, even a rainbow that changes colors!

#### SPECIAL NOTE:

If you do not have the .BMAP files for the exec and intuition libraries, type in the following files using the ED editor. These short files will make a .BMAP file that will ONLY work with this program. Please get an official, complete .BMAP file from your local user group or from AMICUS disk 8, if you plan to explore .BMAPs, or use other programs that need .BMAP files.

```
exec.fd:
##base_SysBase
##bias 198
##public
AllocMem(byteSize,requirements)(D0/D1)
##bias 210
FreeMem(memoryBlock,byteSize)(A1,D0)
##end
```

```
intuition.fd:
##base_IntuitionBase
##bias 60
##public
ClearPointer(Window)(A0)
##bias 270
SetPointer(Window,Pointer,Height,Width,Xoffset,Yoffset)(A
0/A1,D0/D1/D2/D3)
##end
```

Use the ConvertFD program located in the BasicDemos directory of the Extras disk to convert 'exec.fd' to 'exec.bmap' and 'intuition.fd' to 'intuition.bmap'. The file 'graphics.bmap' is already located on the Extras disk in the BasicDemos directory.

**Note:** the resulting .BMAP files should be put in the libs directory of the Workbench disk with the Amiga libraries. You should move the .bmap files onto the same disk as the pointer editor, so the program can access the libraries.

IF YOU PLAN TO EXPERIMENT WITH .BMAP FILES, PLEASE TRY TO GET AN OFFICIAL .BMAP FILE FROM YOUR LOCAL USER GROUP OR AMICUS DISK #8.

#### Program listing:

```

:
: Pointer Image Editor
:
: Copyright 1986 By Stephen R. Pietrowicz
:
: The author places this software in the
: public domain. Please refer to the
:
: Amazing Computing V.1 #6 article
: for more information.
:
: Please do not remove this copyright
: notice.
:
DECLARE FUNCTION AllocMem() LIBRARY
DECLARE FUNCTION FreeMem() LIBRARY
DECLARE FUNCTION ReadPixel() LIBRARY

LIBRARY "exec.library"
LIBRARY "intuition.library"
LIBRARY "graphics.library"
SCREEN 3,320,200,5,1
WINDOW 3,"Pointer Image Editor by SR
Pietrowicz", (0,0)-(309,186),0,3

GOSUB SetUp
:
: Main loop:Wait until the mouse is clicked
: to do anything.
:
Top:
WHILE MOUSE(0) = 0:WEND
x1 = MOUSE(3):y1 = MOUSE(4)
IF (x1 > 143) THEN CheckDial
IF (y1 > 143) THEN CheckColor
x1 = INT(x1/9):y1 = INT(y1/9)
x2 = x1*9+1:y2 = y1*9+1
IF cn = 0 THEN
Ncn = 0
ELSE
Ncn = cn+5
END IF
:
: Set points in both windows, and make sure
: that the "hot spot" stays visible
:
LINE (x2,y2)-(x2+8,y2+8),Ncn,bf
PSET (180+x1,Dial+y1),Ncn
IF (PFlag = 1) AND (x1 = Psx)
AND (y1 = Psy) THEN
LINE (x2+1,y2+1)-(x2+7,y2+7),11,bf
END IF
:
: Set the color that was just set in the
: bitmap
:
IF x1 = 0 THEN
Bit% = &H8000
ELSE
Bit% = (2^(15-x1))
END IF

Sety = Dial-2+y1

IF (cn/2 = INT(cn/2)) THEN
Sa%(Sety,0) = Sa%(Sety,0) AND NOT Bit%
ELSE
Sa%(Sety,0) = Sa%(Sety,0) OR Bit%
END IF
IF (cn < 2) THEN
Sa%(Sety,1) = Sa%(Sety,1) AND NOT Bit%
ELSE
Sa%(Sety,1) = Sa%(Sety,1) OR Bit%
END IF

WHILE MOUSE(0) <> 0:WEND
GOTO Top
:
: Move the dial, and redraw the pointer
: window
:
CheckDial:
IF (x1 < 152) THEN Top
IF (x1 > 163) THEN CheckRGB
MENU OFF
Dbox = Dial
DialTop:
WHILE MOUSE(0) <> 0
Dy = MOUSE(6)
IF (Dy < 2) OR (Dy > 141) THEN DialTop
IF (Dy = Dial) THEN DialTop
LINE (152,Dial)-(163,Dial+2),0,bf
LINE (203,Dial)-(203,Dial+15),0
Dial = Dy

```



```

LINE (152,Dial)-(163,Dial+2),1,bf
LINE (203,Dial)-(203,Dial+15),1
WEND
IF (Dbox = Dial) THEN
  MENU ON
  GOTO Top
END IF

; Redraw the pointer window
PointRedraw:
FOR r = 0 TO 15
  By = r*9+1
  Dly = Dial+r
  FOR s = 0 TO 15
    Bx = s*9+1
    LINE (Bx,By)-(Bx+8,By+8),1,bf
    LINE (Bx,By)-(Bx+8,By+8),
      POINT(180+s,Dly),bf
  NEXT s
NEXT r

; Check to see if the "Hot Spot" goes in
; this window
IF LFlag = 1 THEN
  RETURN
END IF
IF (Psd >= 0) THEN
  IF (ABS(Dial - Psd) >= 0) AND
    (ABS(Dial - Psd) <= 15) THEN
    Psy = Psy-(Dial-Psd)
    Psd = Dial
    IF (Psy >= 0) AND (Psy <= 15)
      THEN
        NPsx = Psx*9+2
        NPsy = Psy*9+2
        LINE (NPsx,NPsy)-
          (NPsx+6,NPsy+6),11,bf
        PFlag = 1
      ELSE
        PFlag = 0
      END IF
    END IF
  END IF
  MENU ON
  GOTO Top

; Change the color that is being used
CheckColor:
IF (y1 < 150) OR (y1 > 170) THEN Top
cn = INT(x1/36)
LINE (225,R1)-(240,R1),0
LINE (255,G1)-(270,G1),0
LINE (285,B1)-(300,B1),0
IF cn = 0 THEN
  LINE (1,176)-(143,184),0,bf
  R1 = 110:G1 = 110:B1 = 110
ELSE
  LINE (1,176)-(143,184),cn+5,bf
  R1 = 110 - (RGB!(cn,1)*100)
  G1 = 110 - (RGB!(cn,2)*100)
  B1 = 110 - (RGB!(cn,3)*100)
END IF
LINE (225,R1)-(240,R1),11
LINE (255,G1)-(270,G1),11
LINE (285,B1)-(300,B1),11
WHILE MOUSE(0) <> 0 :WEND
GOTO Top

; Change the Red, Green, Blue values of the
; current color
CheckRGB:
IF (cn = 0) THEN Top
IF (x1 < 225) OR (x1 > 300) THEN Top
MENU OFF
IF (x1 >= 225) AND (x1 <= 240) THEN Red
IF (x1 >= 255) AND (x1 <= 270) THEN Green
IF (x1 >= 285) AND (x1 <= 300) THEN Blue
MENU OFF
GOTO Top

Red:
WHILE MOUSE(0) <> 0
  Ry = MOUSE(6)
  IF (Ry < 10) OR (Ry > 110) THEN Red
  IF (Ry = R1) THEN Red
  LINE (225,R1)-(240,R1),0
  R1 = Ry
  LINE (225,R1)-(240,R1),11
  RGB!(cn,1) = (110-R1)/100
  PALETTE 5+cn,
  RGB!(cn,1),RGB!(cn,2),RGB!(cn,3)
WEND
GOTO EndRGB

Green:
WHILE MOUSE(0) <> 0
  Gy = MOUSE(6)

```

## A-TALK™

### Advanced Communication and Terminal Program for the AMIGA

- KERMIT - XMODEM - ASCII TRANSFER - Xmodem binary files are stripped of padding characters.
- DIAL-A-TALK - Phone directory, redial and script language for auto-login. Tested login scripts. Programmable function keys.
- ANSI/TTY EMULATION - Resizable and full screen windows. Termcap and terminfo for UNIX users.
- VOICE OPTION - For having mail read aloud and for telling you how the call and login are progressing.
- SETTINGS - Over 10 modem types supported. All communication parameters, including X-on/X-off.

A-TALK lists for \$49.95 and is not copy protected.  
\$2.00 shipping; CA residents add 6.5% sales tax.  
Trade-in discounts available. For info and orders, contact:

Felsina Software  
3175 South Hoover Street, #275  
Los Angeles, CA 90007  
(213) 747-8498

```

IF (Gy < 10) OR (Gy > 110) THEN Green
IF (Gy = G1) THEN Green
LINE (255,G1)-(270,G1),0
G1 = Gy
LINE (255,G1)-(270,G1),11
RGB!(cn,2) = (110-G1)/100
PALETTE 5+cn,
  RGB!(cn,1),RGB!(cn,2),RGB!(cn,3)
WEND
GOTO EndRGB

Blue:
WHILE MOUSE(0) <> 0
  By = MOUSE(6)
  IF (By < 10) OR (By > 110) THEN Blue
  IF (By = B1) THEN Blue
  LINE (285,B1)-(300,B1),0
  B1 = By
  LINE (285,B1)-(300,B1),11
  RGB!(cn,3) = (110-B1)/100
  PALETTE 5+cn,
  RGB!(cn,1),RGB!(cn,2),RGB!(cn,3)
WEND
EndRGB:
MENU ON
GOTO Top

; Initialize data structures and
; variables used by the program
SetUp:
TotalHeight% = 156
DIM Sa$(TotalHeight%,1)
DIM RGB!(3,3)
LFlag = 0

; Memory allocation has to be 4 times
; the height of the pointer image.
; The second parameter to AllocMem()
; must be 2, to allocate memory in
; the first 512K of memory.
MemLength% = TotalHeight% * 4
sia = AllocMem(MemLength%,2%)
IF sia = 0 THEN
  PRINT "Couldn't allocate memory"
  GOTO StopIt
END IF
FOR i = 0 TO TotalHeight%

```



```

LINE (283,9)-(302,111),1,b
LINE (225,110)-(240,110),11
LINE (255,110)-(270,110),11
LINE (285,110)-(300,110),11
R1 = 110:G1 = 110:B1 = 110
LINE (223,113)-(242,123),12,bf
  Sa%(1,0) = 0
  Sa%(1,1) = 0
NEXT i
' Set Up Menus
MENU 1,0,1,"Editor"
MENU 1,1,1,"Load"
MENU 1,2,1,"Save"
MENU 1,3,1,"Clear"
MENU 1,4,1,"Quit"
MENU 2,0,1,"Pointer"
MENU 2,1,1,"Test"
MENU 2,2,1,"Reset"
MENU 2,3,1,"Hot Spot"
MENU 3,0,1,""
MENU 4,0,1,""
PALETTE 30,1,0,0
Psx = 0:Psy = 0:Psd = 2:PFlag = 1
LINE (2,2)-(8,8),11,bf
ON MENU GOSUB CheckMenu
MENU ON
' Pointer drawing box and gadget
LINE (0,0)-(145,145),1,b
LINE (150,0)-(165,145),1,b
Dial = 2
LINE (152,Dial)-(163,Dial+2),1,bf
' Palette that shows how "real" pointer
' looks
LINE (175,0)-(201,158),1,b
LINE (203,Dial)-(203,Dial+15),1
' Draw RGB Settings
PALETTE 12,1,0,0
PALETTE 13,0,1,0
PALETTE 14,0,0,1
LINE (220,0)-(305,158),1,b
LINE (223,9)-(242,111),1,b
LINE (253,9)-(272,111),1,b
LINE (253,113)-(272,123),13,bf
LINE (283,113)-(302,123),14,bf
LOCATE 18,32
PRINT "RGB"
LOCATE 19,30
PRINT "Settings"
' Color Box
PALETTE 6,1,0,0
PALETTE 7,0,1,0
PALETTE 8,0,0,1
RGB!(1,1) = 1:RGB!(1,2) = 0:RGB!(1,3) = 0
RGB!(2,1) = 0:RGB!(2,2) = 1:RGB!(2,3) = 0
RGB!(3,1) = 0:RGB!(3,2) = 0:RGB!(3,3) = 1
LINE (0,150)-(36,170),1,b
FOR i = 1 TO 3
  LINE (i*36,150)-((i+1)*36,170),5+i,bf
NEXT i
LINE (0,175)-(144,185),1,b
' Change the system pointer to the
' default program pointer
DefaultPointer:
RESTORE
' Default program pointer data
DATA 14
DATA -1024,0,30720,-32768,12288,-16384
DATA 6144,-8192,3072,-4096,1536,-10240
DATA 768,-29696,384,1536,192,768
DATA 96,384,48,192,24,96
DATA 12,48,4,24,0,8
rp% = WINDOW(7)
READ Ap%
POKEW si%+0
POKEW si%+2,0
Padd = 4
FOR i = 1 TO (Ap%+1)*2
  READ pl%
  POKEW si%+Padd, pl%

```

```

Padd = Padd + 2
NEXT i
POKEW si%+Padd, 0
POKEW si%+Padd+2, 0
PALETTE 17,1,0,0
PALETTE 18,.6,0,0
PALETTE 19,0,.6,.8
HotX% = 0
HotY% = 0
CALL SetPointer(rp%, si%, Ap%+1,16,HotX%,HotY%)
RETURN
' Menu functions
CheckMenu:
id = MENU(0)
item = MENU(1)
MENU OFF
' Editor
IF id = 1 THEN
' Load pointer from a file
IF item = 1 THEN
  FileName$ = ""
  GOSUB GetFileName
  IF FileName$ = "" THEN LoadDone
  GOSUB ClearImage
  OPEN FileName$ FOR INPUT AS #1
  INPUT
  #1,RGB!(1,1),RGB!(1,2),RGB!(1,3)
  INPUT
  #1,RGB!(2,1),RGB!(2,2),RGB!(2,3)
  INPUT
  #1,RGB!(3,1),RGB!(3,2),RGB!(3,3)
  INPUT #1,Ap%
  FOR j = 0 TO Ap%
    INPUT #1,Sa%(j,0),Sa%(j,1)
  NEXT j
  INPUT #1,Psx, Psy
  CLOSE #1
  Psx = -Psx
  Psy = -Psy
  Psd = INT(Psy/16)*16+2
  Psy = Psy-Psd+2
  PALETTE
  6,RGB!(1,1),RGB!(1,2),RGB!(1,3)
  PALETTE
  7,RGB!(2,1),RGB!(2,2),RGB!(2,3)
  PALETTE
  8,RGB!(3,1),RGB!(3,2),RGB!(3,3)
  IF (Psd+Psy <= 15) THEN
    PFlag = 1
  ELSE
    PFlag = 0
  END IF
' Reconstruct the colors, and draw them in
' the palette
FOR Scan = 0 TO Ap%
  Bit% = &H8000
  Bit0% = (Sa%(Scan,0) AND &H8000)/&H8000
  Bit1% = (Sa%(Scan,1) AND &H8000)/&H8000
  cn = (Bit1%*2) OR Bit0%
  IF cn = 0 THEN
    Ncn = 0
  ELSE
    Ncn = cn+5
  END IF
  PSET(180,Scan+2),Ncn
  FOR j = 14 TO 0 STEP -1
    Bit% = (2^j)
    Bit0% = (Sa%(Scan,0) AND Bit%)/Bit%
    Bit1% = (Sa%(Scan,1) AND Bit%)/Bit%
    cn = (Bit1%*2) OR Bit0%
    IF cn = 0 THEN
      Ncn = 0
    ELSE
      Ncn = cn+5
    END IF
    PSET(195-j,Scan+2),1
    PSET(195-j,Scan+2),Ncn
  NEXT j

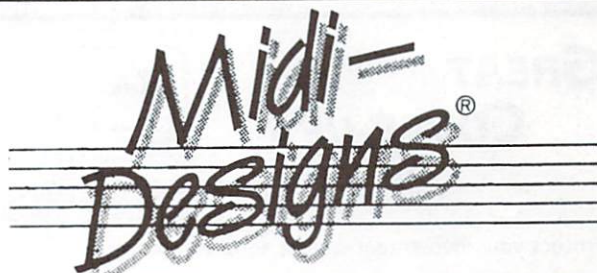
```



```

NEXT Scan
: Reconstruct the main drawing area
: and reset the intuition pointer
:
LFlag = 1
GOSUB PointRedraw
IF (PFlag = 1) THEN
    LINE (Pxx*9+2,Psy*9+2)-(
        Pxx*9+8,Psy*9+8),11,bf
END IF
LFlag = 0
LoadDone:
MENU ON
RETURN
END IF
: Save current pointer to a file
:
IF item = 2 THEN
: Make sure there is a pointer to save...
    Ap% = 156
    PFlag = 1
    WHILE (PFlag = 1)
        IF (Ap% < 0) THEN
            PFlag = 0
        ELSEIF (Sa%(Ap%,0) <> 0) OR
            (Sa%(Ap%,1) <> 0) THEN
            PFlag = 0
        ELSE
            Ap% = Ap% - 1
        END IF
    WEND
    IF (Ap% < 0) THEN
        LOCATE 21,20
        PRINT "No pointer! Hit ESC "
        GOTO SaveBad
    END IF
save    FileName$ = ""
    GOSUB GetFileName
    IF FileName$ = "" THEN SaveDone
    OPEN FileName$ FOR OUTPUT AS #1
    WRITE
    #1,RGB!(1,1),RGB!(1,2),RGB!(1,3)
    WRITE
    #1,RGB!(2,1),RGB!(2,2),RGB!(2,3)
    WRITE
    #1,RGB!(3,1),RGB!(3,2),RGB!(3,3)
    WRITE #1,Ap%
    FOR j = 0 TO Ap%
        WRITE #1,Sa%(j,0),Sa%(j,1)
    NEXT j
    WRITE #1,-Pxx, -(Psy+Psd-2)
    CLOSE #1
    GOTO SaveDone
SaveBad:
    Key$ = INKEY$: IF Key$ = "" THEN
SaveBad
    IF ASC(Key$) <> 27 THEN SaveBad
    LOCATE 21,20
    PRINT "
SaveDone:
    MENU ON
    RETURN
    END IF
: Clear current pointer bitmap and drawing
: areas
:
IF item = 3 THEN
ClearImage:
    FOR i = 0 TO 156
        Sa%(i,0) = 0
        Sa%(i,1) = 0
    NEXT i
    LINE (1,1)-(144,144),0,bf
    LINE (152,Dial)-(163,Dial+2),0,bf
    LINE (152,2)-(163,4),1,bf
    LINE (203,Dial)-(203,Dial+15),0
    Dial = 2
    LINE (176,1)-(200,157),0,bf
    LINE (203,2)-(203,17),1
    Pxx = 0:Psy = 0:Psd = 2:PFlag = 1
    LINE (2,2)-(8,8),11,bf
    MENU ON
    RETURN
    END IF
: Quit
:
IF item = 4 THEN
    GOTO StopIt

```



## MIDI IS HERE!

Midi-Designs Is Proud To Introduce The  
MD-1 Midi Interface For The Commodore Amiga.  
Features Include:

- IN, OUT, THROUGH jacks for sending and receiving data
- Attractive custom metal enclosure
- 1 year warranty
- 100% compatible with Activision's Music Studio™ Software and all popular synthesizers including those manufactured by Roland, Yamaha, Casio and Korg.

PRICE: **\$49.95**

MD-1 INTERFACES ARE IN STOCK FOR IMMEDIATE SHIPMENT  
For Additional Information or to Order, write or call:

MIDI-DESIGNS  
2232 Summit Street  
Columbus, OH 43201  
(614) 267-6755  
Dealer Inquiries Invited

Midi-Designs is a division of J. Michaels Company

```

END IF
END IF
: Pointer functions
IF id = 2 THEN
: Use the current pointer image
    IF item = 1 THEN
        GOSUB SetIt
        MENU ON
        RETURN
    END IF
: Reset the pointer in use to the default
: pointer
    IF item = 2 THEN
        GOSUB DefaultPointer
        MENU ON
        RETURN
    END IF
: Set the pointer's "Hot Spot"
    IF item = 3 THEN
PickSet:
        WHILE MOUSE(0) = 0:WEND
        x1 = MOUSE(3):y1 = MOUSE(4)
        IF x1>143 OR y1>143 THEN PickSet
        x1 = INT(x1/9):y1 = INT(y1/9)
        x2 = x1*9+2: y2 = y1*9+2
        IF (PFlag = 1) THEN
            NPsx = Pxx*9+2
            NPsy = Psy*9+2
            Pc = POINT(NPsx-1,NPsy-1)
            IF (Pc <> 0) THEN
                LINE (NPsx,NPsy)-
                    (NPsx+6,NPsy+6),Pc,bf
            ELSE
                LINE (NPsx,NPsy)-
                    (NPsx+6,NPsy+6),0,bf
            END IF
        END IF
        LINE (x2,y2)-(x2+6,y2+6),11,bf
        Pxx = x1
        Psy = y1
        Psd = Dial
    END IF

```



# GREAT COVER-UPS



Protect your investment with opaque vinyl covers.

|                                  |        |
|----------------------------------|--------|
| Amiga and Monitor                | \$8.95 |
| Printers                         | \$4.95 |
| (Specify brand, model and width) |        |
| 3 1/2" Disk Drive                | \$1.95 |
| 5 1/4" Disk Drive                | \$1.95 |
| Sidecar                          | \$3.95 |

TO: GREAT COVER-UPS Phone: (503) 246-8977  
6805 SW 8th Avenue  
Portland, Oregon 97219

SEND ME:

\_\_\_\_\_ Amiga & Monitor Covers @ 8.95 ea  
\_\_\_\_\_ @ \_\_\_\_\_ ea  
\_\_\_\_\_ @ \_\_\_\_\_ ea

Please add \$1.25 each for postage and handling  
Dealer inquiries invited

```

PFlag = 1
WHILE MOUSE(0) <> 0:WEND
ON MENU GOSUB CheckMenu
MENU ON
RETURN
END IF
END IF
RETURN
END
' Subroutine to return FileName$
' FileName$ is limited to 17 characters
' Hitting the escape key exits with a
' NULL value
GetFileName:
LOCATE 21,20
PRINT "Input file Name:"
LINE(150,173)-(300,185),1,b

Key$ = INKEY$
WHILE Key$<>"":Key$ = INKEY$:WEND

Box = 152
LINE(Box,175)-(Box+7,183),30,bf
Cursor = 20
LOCATE 23,Cursor

NameTop:
Key$ = INKEY$:IF Key$ = "" THEN NameTop
NameLen = LEN(FileName$)
IF (ASC(Key$) = 27) THEN
    FileName$ = ""
    GOTO NameDone
END IF
IF (ASC(Key$) = 13) AND
(NameLen <> 0) THEN NameDone
IF (ASC(Key$) = 8) AND (NameLen > 0)
THEN
    FileName$ = LEFT$(FileName$,NameLen-1)
    LINE (Box,175)-(Box+7,183),0,bf
    Box = Box-8
    LINE (Box,175)-(Box+7,183),30,bf
    GOTO NameTop
END IF
IF (NameLen >= 17) THEN NameTop
IF ((Key$ >= "0") AND (Key$ <= "9")) THEN

```

```

NameAdd
IF ((Key$ >= "A") AND (Key$ <= "Z")) THEN
NameAdd
IF ((Key$ >= "a") AND (Key$ <= "z")) THEN
NameAdd
GOTO NameTop
NameAdd:
FileName$ = FileName$ + Key$
LINE (Box,175)-(Box+7,183),0,bf
LOCATE 23,20+NameLen
PRINT Key$:
Box = Box+8
LINE (Box,175)-(Box+7,183),30,bf
GOTO NameTop

NameDone:
LOCATE 21,20
PRINT " "
LINE(150,173)-(300,185),0,bf
RETURN
END
' User the pointer image on the palette as
' the default pointer
SetIt:
Ap% = 156
PSFlag = 1
WHILE (PSFlag = 1)
    IF (Ap% < 0) THEN
        PSFlag = 0
    ELSEIF (Sa%(Ap%,0) <> 0) OR
        (Sa%(Ap%,1) <> 0)
    THEN
        PSFlag = 0
    ELSE
        Ap% = Ap% - 1
    END IF
WEND
IF (Ap% < 0) THEN
    RETURN
END IF

POKEW si&, 0
POKEW si&+2, 0

Padd = 4
FOR j = 0 TO Ap%+1
    POKEW (si&+Padd), Sa%(j,0)
    Padd = Padd + 2
    POKEW (si&+Padd), Sa%(j,1)
    Padd = Padd + 2
NEXT j
POKEW si&+Padd, 0
POKEW si&+Padd+2, 0

PALETTE 17,RGB!(1,1),RGB!(1,2),RGB!(1,3)
PALETTE 18,RGB!(2,1),RGB!(2,2),RGB!(2,3)
PALETTE 19,RGB!(3,1),RGB!(3,2),RGB!(3,3)

Spx% = -Psx: Spy% = -(Psy+Psd-2)
CALL SetPointer(rp&, si&,
Ap%+2,16,Spx%,Spy%)
RETURN
' Clean up the loose ends, and exit
StopIt:
WINDOW CLOSE 3
SCREEN CLOSE 3

CALL FreeMem(si&,MemLength%)
LIBRARY CLOSE
END

```

•AC•



# AmigaNotes

By Richard Rae  
Music Editor  
CIS [72177,3516]

I am delighted to announce that Amiga music is a reality! The AmigaNotes Award for First Musical Software goes to ActiVision for their Music Studio, which began appearing on store shelves in mid May; the hardware nod goes to Golden Hawk Technologies for their MIDI Gold interface. I will be doing an in-depth comparative review on MIDI Gold at a later date; suffice for now to say that it is nicely constructed and functions well.

This month's column is devoted to an in-depth review of Music Studio. Before I delve into that subject, however, I should explain my reviewing style. When looking at a new product I tend to be very critical, and decide on personal purchases based on which item has the least number of features I don't like. Even the rare gushingly enthusiastic review will contain a few points of contention. The idea here is not to pan the product, but to present a fair appraisal so that YOU can decide which package best suits your needs.

And you always have the rebuttal: if you think I've been too harsh in any of my comments, drop me some Email and let's discuss it.

## REVIEW: ACTIVISION MUSIC STUDIO

ActiVision bills Music Studio as "(a program) so complete it... offer(s) every option and audio function anyone could possibly want". As is typical of systems which attempt to do everything, Music Studio is a jack of all trades and master of none. The designers (Audio Light) did a good job of balancing capabilities with limitations, though; the result is a reasonably well thought out package which many will find useful in one capacity or another.

First let's take a look at what Music Studio isn't. It is not a MIDI sequencer/recorder, though it does provide very limited MIDI support. It is not a professional scorewriting package, yet it gives you the capability to print sheet music. And it is not the ultimate demonstration of the Amiga's sound producing capability, although its "synthesizer" does provide four simultaneous voices and quite a bit of control over the sounds produced.

Retailing at \$59.95, Music Studio comes in a colorful little box containing the program diskette, 78 page instruction booklet, icon reference card, and an ActiVision product flyer. The manual covers both Amiga and Atari ST versions of the program. This provides an interesting opportunity to contrast the capabilities of the two versions and, to a small extent, the relative power of the two machines.

The disk is copy protected, but the scheme is fairly straightforward: a bad sector header which causes DISKCOPY to fail with a source error. Since COPY moves

individual files, COPY DF0: TO DF1: ALL will copy the disk (albeit slowly). You won't be able to boot from this backup, however: the program includes a programmed check which looks for the bad sector, and you'll enjoy a visit from the Guru if it isn't found.

The backup is still useful if you treat it like a key disk protection scheme: boot from the master copy, then put it away and run from the backup. In fact, you can load your backup copy into drive zero and the master in drive one for booting: when the program checks for the disk protection, it will try all drives in sequence until it finds it. This allows you to copy Music Studio to a hard disk and insert the master only for booting. Marauder will successfully create a bootable copy, and this is probably the best solution for floppy based systems.

Music Studio will run on a 512K single drive system, although a second drive makes life considerably more pleasant. With the exception of the bad sector header, the program disk is standard AmigaDos format and boots directly from 1.1 KickStart. It comes up with a cute introductory screen and (bypassable) theme music which will immediately give you an idea of what the package is capable of. Do yourself a favor: run the audio lines to a good stereo! Your monitor speakers just won't cut it when sound is of primary importance.

Included on the disk are many demo songs, some for Amiga voices, others for an external Casio CZ101 synthesizer. Interestingly enough, some of the best songs are hidden from view unless you change the default directory match pattern. Be sure to listen to anything with a .song or .sng extension anywhere on the disk, or you'll miss some good material. (One of my favorites among the hidden is Chris French's "AhWishAHadAModem".)

## COMPOSITION AND PLAYBACK

The composing section of Music Studio supports two approaches to creating music. For the traditionalists among us, the main composing screen allows placement of notes on a standard musical staff. For avant-garde composers or non-musicians, a "musical paintbox" mode is available. The standard notation screen provides treble and bass staves on which to place 32nd to whole notes and 32nd to whole rests. In addition, any note may be dotted (extending its duration by 50%) or tied to other note(s), and triplets are supported. All major keys are available, and any note may be placed with an accidental to depart from the selected key's scale. Eight time signatures ranging from 2/2 to 6/8 are available for your compositions.

Several block oriented editing functions are provided which simplify score entry and modification. These functions include making space between already placed notes to make additions, moving or copying a block of music, changing instruments, shortening or lengthening note and rest durations, transposing up or down a step at a time, and adding repeat bars. These functions operate on the entire



## GETTING YOUR MONEY'S WORTH?

### SKE

A COMPANY DEDICATED TO PRODUCING  
QUALITY SOFTWARE AT  
AFFORDABLE PRICES, LIKE:

#### SKETerm

- UP TO 19200 BAUD
- TELEPHONE DIRECTORY
- USER DEFAULTS & PARAMETERS
- ASCII, XMODEM & KERMIT FILE XFERS
- VT100/ADM3A/ANSI/TTY/D200/OTHERS
- USER DEFINED XON-XOFF
- ON-LINE HELP
- HAYES MODEM SUPPORT
- SUPPORTS BACKGROUND FILE XFER
- MULTITASKING & WRITTEN IN C

**PRICE \$49.95**

PLUS \$2.50 SHIPPING (FLA. RES ADD 5%) VISA/MC/  
COD/CHK ACCEPTED CALL 813-787-3111 TO ORDER  
TODAY OR WRITE TO:

**SKE Software Co.**  
**2780 COTTONWOOD COURT**  
**CLEARWATER, FL. 33519**  
**(813) 787-3111**

song or an area which has been previously "marked". In operation this is very similar to editors supporting the "cut and paste" mode. Moving about in a composition longer than the screen is accomplished either by repeatedly clicking left and right arrows or by moving a slider bar near the bottom of the screen. You can insert up to three verses of lyrics in addition to the music, but be advised that this uses memory normally reserved for notes. Without lyrics the storage limit is approximately 8000 notes and rests.

The "Music Paintbox" offers an interesting alternative to standard notation. Instead of note symbols, musical tones are represented by rectangles, the lengths of which correspond to note durations. Your composition may be started on either screen and you can flip between them with a single mouse click.

Both composing modes support a feature which allows you to assign various instruments to each of four virtual tracks.

By turning the individual tracks on and off, you can hear any portion of your composition in isolation. For example, you can work on a horn part by turning off the bass guitar and piano parts. This is also useful when scoring for MIDI, where you might have more sound sources than the Amiga alone: you could compose up to four parts at once using only the computer's output, then turn on all the tracks for the actual performance.

Playback functions include a tempo control which extends from 57 to 200 beats per minute in 30 steps, and a volume slider which covers the musical range from pianissimo to

fortissimo. Two playback modes let you select whether the musical score scrolls in time with the music or remains fixed.

During the construction of a score you might want a hardcopy on which to jot notes or ideas. Music Studio has a function to print a song, but unfortunately it only supports Epson compatible printers and does not appear to vector through Preferences. Considering the Epson's market penetration and pricing schedule, though, they couldn't have picked a much more universal machine.

#### INSTRUMENT CREATION

An instrument is Audio Light's name for a sound used in a composition. You may define up to fifteen instruments in a bank, and all of them may be used in a single composition if you wish, although only four may sound simultaneously. Each instrument is identified by one of fifteen unique colors, and notes or paintbox rectangles are displayed in the color of the instrument which will play them. This is a cute idea, since each instrument would be a different timbre, which is often referred to as "tone color".

An instrument is built up from component parts called harmonics, and you have complete control of how they vary over time. (For those not familiar with the concept of harmonics and envelopes, Amazing Computing will be running a tutorial on sound synthesis in the future. Watch this space for details, and in the meantime, experimentation is the best teacher!)

Music Studio provides you with 32 sound sources: the first 31 harmonics of a tone plus what Audio Light calls "noise".

This is not noise in the traditional sense, but a harsh, raspy tone at the fundamental frequency. Up to seven of these 32 sound sources may be selected for creating any instrument. Each of the seven selected harmonics has an associated envelope describing its changes over time. The manual indicates that the maximum non-paused envelope time is three seconds, but in reality it is just shy of six. Up to seven points may be marked within this time period with a resolution of 20 milliseconds. These marked points are then used as end points for the segments of the harmonic envelopes. In this manner it is possible to create relatively complex sounds with up to six envelope segments per harmonic.

Normally, the program cycles through the envelope and ends the tone in the specified time regardless of actual note duration. By flagging any of the marked points as a sustain point, the tone can be made to pause at that location until the note expires, at which time the remainder of the envelope is used. The former mode is typical of piano, guitar, and percussive instruments, while the latter is good for simulating organ, violins, and wind instruments.

Other instrument parameters include tremolo (amplitude modulation) and vibrato (frequency modulation). There are no provisions for introducing a delay or changing the modulating frequency. It is also possible to assign an instrument to the left channel, the right channel, both channels (center), or "none of the above".

The last mode is interesting in that an instrument is assigned to whatever DAC channel is currently available, resulting in



voices jumping from side to side during a composition. This can add a wonderful liveliness to a piece. As an aside, you should understand that assigning an instrument to the center of the stereo field by using two channels cuts the maximum number of simultaneous notes to three. If your piece calls for a center-stage trumpet, you can only add a bass and piano on left and right sides. By assigning the trumpet to one side instead you can add a fourth instrument such as a guitar.

Once you are satisfied with the instrument sounds in the current bank, you may save the bank of instruments in a sound file for use by several songs. Or, you can forget about the instruments and get on with writing your composition: when the song is saved the bank of instruments for that song are automatically saved as well. This means that you can load a song and play it immediately without having to fumble about looking for the correct bank of instruments to go with it.

When loading a song or instrument bank, or creating a new sound, the Amiga will pause while it recalculates the sound data. Depending on how many instruments need to be updated, this can take upwards of 20 seconds. I had no problem with this short delay, but some people I have talked to find it a minor irritation.

## MIDI SUPPORT

With an optional MIDI converter Music Studio can control external devices. Several interfaces, ranging in price from fifty to several hundred dollars, will be available within a few months; Music Studio supports any interface which connects to the serial port. (For those of you unfamiliar with MIDI, Amazing Computing will also be running a MIDI tutorial in the months ahead.)

Music Studio views external MIDI devices as additional sound sources only. You cannot "compose at the keyboard", use Music Studio to record a song in real time, or remotely change parameters on the synthesizer. Using the MIDI output simply increases the maximum number of simultaneous notes, with roughly the same amount of control inherent in the internal voices.

Upon first glance at the manual, you may be disappointed to find that the Atari version allows you to compose a song from a MIDI keyboard while the Amiga version does not. But read on; we're not missing much! The MIDI keyboard selects ONLY the note itself; you must still use the mouse to select the note durations and starting location. (Personally, this mode hardly seems worth the effort it must have taken to implement on the Atari.)

The MIDI screen allows you to name each of the fifteen instruments and assign a MIDI channel, preset, and key window to each. By selecting different channels for the instruments you can drive several synthesizers and perhaps a drum machine, with each unit playing independently of the others. Or, if you have a multitimbral instrument such as those in Casio's CZ series, you can play several different instrument sounds simultaneously with one synthesizer. The



# LIONHEART

## BUSINESS & STATISTICAL SOFTWARE

PC/MS-DOS (5<sup>1</sup>/<sub>4</sub> & 3<sup>1</sup>/<sub>2</sub>), MACINTOSH, AM'GA  
ATARI ST, C64/128, CP/M, APPLE DOS 3.3

Explanatory books with professional compiled software; the new standard for statistical use. The influential Seybold Report on Professional Computing has this to say about Lionheart "... our sentimental favorite because of its pragmatic approach to the basic statistical concepts... The thinking is that the computer merely facilitates the calculations; the important thing is to be able to formulate a problem correctly and to determine what type of analysis will be most valuable." Let Lionheart help you get ahead of the competition! Spreadsheet compatible.

|                                                     |       |
|-----------------------------------------------------|-------|
| • BUSINESS STATISTICS.....                          | \$145 |
| • EXPERIMENTAL STATISTICS .....                     | 145   |
| • MULTIVARIATE ANALYSIS .....                       | 150   |
| • EXPLORATORY DATA ANALYSIS .....                   | 75    |
| • STATISTICS FOR MARKETING .....                    | 150   |
| • QUALITY CONTROL & INDUSTRIAL<br>EXPERIMENTS ..... | 145   |
| • FORECASTING AND TIME-SERIES .....                 | 145   |
| • SALES AND MARKET FORECASTING .....                | 145   |
| • DECISION ANALYSIS TECHNIQUES .....                | 110   |
| • LINEAR & NON-LINEAR PROGRAMMING .....             | 95    |
| • PERT & CRITICAL PATH TECHNIQUES .....             | 95    |
| • OPTIMIZATION .....                                | 110   |

VISA, MasterCard, AMEX, Check

P.O. Box 379, ALBURG, VT 05440

(514) 933-4918

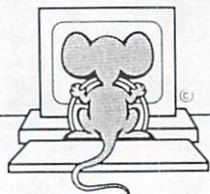
manual mentions that only one note may be played via each MIDI channel. This is a misleading statement, as it applies only to certain instruments driven in certain modes. For many applications you can simultaneously play as many notes as your synthesizer has voices.

For synthesizers with multiple stored sounds, the preset column allows Music Studio to automatically select the correct timbre. Thus you can set two different instruments, used in different parts of the song, to the same MIDI channel with different preset values, and the synthesizer responding to that channel will change from strings to horns to piano and back without any user intervention. There is a limitation to this mode, however: for some reason it will not operate in conjunction with the Amiga's internal voices. If you play your song from the MIDI function screen (which disables the internal voices), or use the options menu to turn the Amiga sound off, Music Studio will properly select the desired programs on the external devices. With the Amiga voices enabled, however, the synthesizer is never sent a program change.

The key window defines a five octave range over which each instrument will be played. This feature allows you to scale synthesizer voices to properly match each other and the voices from the Amiga. For example, if you have a cello sound which you manually play around the middle of your synthesizer keyboard, it may be too low when Music Studio drives it with notes on the bass clef. Key windowing allows you to transpose the sound up an octave or so without having to rewrite the score or reprogram the sound.



# Mouse Driven



Classic games software you can drive with your mouse! But, you don't need a license -just an AMIGA and:

## ™ Games Gallery I, II, and III.

Each of these packages contain exciting:  
**Space, Gambling, Sports Games, and Mind Teasers.**

Each provides a standard series of features and options for:

•Speech •Graphics •Menus  
•Color •Help •Voice and •Mouse Control!

Kickstart 1.1 & 512K memory required. \$29.95  
+ \$3.00 shipping & handling.

(713) 488-2144

Telephone orders  
welcome  
Visa Mastercard Amex

**MSI MERIDIAN™  
SOFTWARE  
INC.**  
P.O. Box 890408  
Houston, TX. 77289-0408  
AMIGA is a trademark of Commodore-Amiga, Inc.

Music Studio allows you to use any mix of internal and MIDI sounds in your compositions. Menu selections allow you to toggle both the Amiga and MIDI outputs on and off. By running both outputs simultaneously you can "layer" sounds; for example, the Amiga can provide chimes while your synthesizer plays the same notes as a string ensemble. With careful setup you can gain even more control. Individual instruments can be muted in the Amiga by defining "null" voices which make no sound; these instruments can be assigned to an external synthesizer to be played. Conversely, internal Amiga instruments can be assigned to unused MIDI channels to silence the external devices. By combining this technique with layering, some very impressive arrangements can be developed.

As with the instruments, the MIDI parameters are stored along with the song, simplifying later playback. You will have to turn the MIDI mode on manually from the options menu, however. (A quick plug for MIDI: The Amiga has a great deal of audio potential, but if you are serious about making music it should be but one tool in your arsenal. Synthesizers don't have to be expensive; as an example, the Music Studio manual and demo songs are geared towards the Casio CZ101. This is an 8 voice multitimbral MIDI synthesizer which has been sold for less than \$250.)

### PEEVES

Now comes the point at which I start railing about all the little things which irritated me while reviewing Music Studio.

Manufacturers, can't we please give up on copy protection? The only ones who truly benefit from CP are the writers and marketers of copy buster programs.

Second, please, PLEASE, PLEASE write protect your disks before you ship them! Like almost all software I receive, the Music Studio diskette is shipped with the write protect tab in the enabled position. This is dangerous under normal circumstances, but potentially fatal with CP'd software.

People, save yourself some grief: flip that tab to the inhibit position before you insert it in the drive for the first time. Better yet, use a pair of needle-nose pliers to REMOVE the tab from all your master diskettes. (If you do happen to munge your only copy of Music Studio, Activision will replace it free during the first 90 days, and for \$7.50 thereafter.)

There is no index in the owner's manual. A detailed table of contents is nice, but when I'm actually trying to USE a package I live by the index. Think I'm exaggerating the problem? Try to find the section on using repeats. Really.

While on the subject of the manual, I should mention that I spotted very few errors; Activision seems to have fairly good proofreaders. The only irritating mistake was related to deleting ties. The manual indicates this is done by clicking once over the tie with a new tie. In reality, you must place a new tie over the old one; in other words it takes two clicks: one on the first note of the tie, the other on the last note. Took a while to figure that one out!

For some reason Audio Light chose to depart from the Amiga mouse interface standard: both mouse buttons do exactly the same thing! Rather than pulling up the menu bar, the right button duplicates the "pick" function of the left button. This is strange, especially when one learns that the Atari version assigns different functions to each button. Fortunately the "one button mouse" is not too difficult to get used to, but it is an unnecessary departure from the norm.

Also relating to the user interface, Audio Light could have handled the file requesters a bit better. When you need to deal with a song or sound file, Music Studio puts up a list of the current file names along with a manual entry area; you can either use the mouse to pick your file from the list or enter it directly. Trouble is, you always have to wait for the list to be posted even if you don't want or need it. When saving a new file the list does you absolutely no good, but you'll still cool your heels for a minute or so while the drive gronks its little heart out. The ability to click in the manual entry field and enter a file name immediately would have really been nice; others have done it, so we know it's possible.

I wish the programmers had picked a different color palette. The staffs are black on a brilliant white background. This tends to make the screen extremely bright, and renders the yellow instrument almost unusable because the notes blend into the background. Turn the brightness and contrast down to a more comfortable level and the black, gray, brown, and dark green notes all start to look the same. I think a black or dark gray background would have avoided this problem.



Since there is no way to exit the program, there is no way to officially change Preferences. This is unfortunate, at least for me; I prefer a peppier mouse than the setting they provided. In fact, I require it: Music Studio's default setting requires about half a foot more table space than I can devote to my mouse. While reviewing this program I was constantly picking the little critter up and moving him to the other side of his workspace to continue a cursor action. Or at least until I got tired enough of the situation to do something about it.

On the Amiga, Preferences settings are always stored in a file called 'system-configuration' in the 'devs' directory. By copying this file to another disk, you immediately set up your environment without going through Preferences. If you make a backup copy of Music Studio as described earlier, drop it in drive one while your system disk is in drive zero, and type

```
COPY    DF0:DEVS/SYSTEM-CONFIGURATION    TO
DF1:DEVS/SYSTEM-CONFIGURATION
```

you will replace Music Studio's Preference settings with your own. If you then boot from the backup with the master in drive one, Music Studio will use your settings. (Of course, if you're really gutsy you could try to replace the file on the master disk, but I do NOT recommend this. Ideally you should never write to ANY master disk.) This little trick will not change the palette Music Studio uses (they set up their own inside the program), but it will let you center the screen, change the sensitivity of your mouse, and even use your customized pointer.

This is a multi-tasking machine by nature, but Music Studio doesn't acknowledge that fact. Once you begin playback of a piece, you can do exactly nothing. You can't change volume, you can't change tempo, you can't change instruments. Well... to be perfectly fair you CAN do one thing: you can stop playback. But that's all! The system is obviously capable of more: eject and reinsert a disk while a composition is playing, and you will hear the disk being validated while the score plays on, never missing a beat. It would have been nice to be able to change the tempo, at least.

If you intend to do professional quality recording, you should plan to rely on external synthesizers driven by Music Studio's MIDI, as the internal voices are hardly in the DX7 class. Audio Light seems to have used fairly large amplitude steps in their program, and the result is a sort of "roughness" during slow attacks and releases. Also, when a fifth sound is needed the program "yanks" a DAC away from a currently sounding note, with no concern for providing a smooth decay. The result can be random clicks and pops as DACs are reassigned.

I have lots of problems with the musical notation used. Minor nits include the fact that measure bars must be placed manually and may be placed incorrectly, and printed staves are split wherever convenient instead of at the end of a measure.

More serious difficulties abound: there are no downward stems, no stacked notes, no beams, and only one ledger line

# MIDI GOLD

MIDI Interface for the Commodore Amiga Personal Computer  
Available now...from Golden Hawk Technology!

## FEATURES:

- Dual MIDI Out and single MIDI In connections.
- Sync Out connection provides clock and start/stop control for drum machines and other devices.
- Connects directly to the serial port.
- Custom metal enclosure.
- Complete with interface cable.
- One year warranty.
- Full technical information included.

**PRICE \$79.00** - Free shipping on prepaid orders.

## Golden Hawk Technology

427-3 Amherst Street, Suite 389, Nashua, NH 03063  
(603) 882-7198 Weekdays 2PM - 10PM EST

VISA/MasterCard/CODs Accepted

DEALERS INQUIRIES WELCOME

Amiga is a trademark of Commodore-Amiga, Inc.

is displayed per note regardless of how far above or below the staff it is. Thinking that perhaps I was being too critical, I showed a copy of a Music Studio printout to two musicians, one a semi-professional organist and the other a local rock artist. Both said that they would have to do a lot of cleanup, such as inserting ledger lines, before they could play the song I handed them. Remember that the appearance of the score has absolutely **nothing** to do with Music Studio's ability to play the music. If you're only interested in dumping a hardcopy of your score for reference or to make notes on, these limitations probably won't bother you. But don't expect to be able to easily read a composition printed by Music Studio unless it is very simple.

An important omission on the composing screen is control over dynamics: there is no way to make notes louder or softer within the piece. Music Studio supposedly supports an accent, and you can indeed place accented notes. But they don't get accented! The manual says any accented notes are played fortissimo, but there is no change in volume for either the internal voices or the MIDI channels.

In composing or transcribing music, you will quickly discover a frustrating fact: the score is based on "columns", not notes, and some durations can be misinterpreted if this is not taken into account. As an example, write a measure containing four quarter notes, then place an eighth note above the first and third quarter notes. The first quarter and eighth note will sound simultaneously. When the eighth note expires, Music Studio will immediately move to the next "column", cutting the first quarter note to one half of its



intended duration and destroying the timing of the measure. The same thing will happen to the third quarter note.

The workaround is to place a rest in the column after each eighth note, but according to the rules of traditional notation this should not be necessary. The upshot of this is that transcribing a piece of sheet music for Music Studio is more difficult than it should be: if you enter an exact copy of the printed score, odds are it will not play correctly and will require numerous rests scattered throughout the piece. In fact, several of the demo songs included with Music Studio contain a continuous string of rests running along the top edge of the score; I can't help but wonder if they are there to patch around this problem.

Remember that I've picked nits here in some cases. There is much good about Music Studio, and overall I like the program. I can't say it will become my most used composition program, but neither can I point to something better. Activision has a very big plus in their favor: they were FIRST with a real music program for the Amiga. For the price, you can afford to be wrong.

#### ACKNOWLEDGMENTS

I'd like to thank Charlotte Taylor at Activision for getting my copy of Music Studio into one of the first shipments so I could do a timely review. Thanks also to Jeff Arnold at Golden Hawk Technologies for rushing me one of his MIDI interfaces (to use in a review of somebody else's product!).

A tip of the hat to the Hermitage Road ComputerLand in Richmond, Virginia, for the loan of an Epson printer (they don't sell Amigas, but they are very accommodating folks.) And I owe a debt of gratitude to Chuck and Don at Don Warner Music, also in Richmond, for the loan of a CZ101. (I don't have one of those in my keyboard stack).

That's all for now. 'Til next month...

Nybbles, Rick

#### SUMMARY: ACTIVISION MUSIC STUDIO

My overall impression of Music Studio is favorable. There are some serious limitations and a few rough edges, but we need to keep in mind that this is the first musically oriented package ever released for the Amiga. If you are interested in experimenting with music and don't need professional quality scoring or sound, Music Studio is definitely worth considering.

PRODUCT: Music Studio \$59.95

COPY PROTECTION: Yes, bypassable as "key disk"

REQUIREMENTS: Amiga with 512K, one drive, KS 1.1

ActiVision

P.O. Box 7287

Mountain View, CA 94039

415-960-0410

**Amiga™ Dreams come true with EASYL™**

USE WITH:  
IMAGES™ / ANIMATOR™  
GRAPHICRAFT™  
DELUXE PAINT™  
OTHERS (Source code included)

APPLICATIONS:  
-ANIMATION  
-A.V. PRESENTATIONS  
-HOME  
-EDUCATION  
-GRAPHIC ARTS  
-FINE ART  
-VIDEO COLORIZATION

**\$499**

TO ORDER, PLEASE CALL:  
**1-416-744-4246**  
Anakin Research Inc.  
100 Westmore Drive, Unit 11C,  
Rexdale, Ontario, Canada M9V 5C3

Amiga™, Images™/Animator™, Graphicraft™, Deluxe Paint™, Easy™, Acme™, Oxberry™, and IBM™ are trademarks of Commodore/Amiga, Aegis Development, Commodore/Amiga, Electronic Arts, Anakin Research Inc., Acme, Oxberry and IBM respectively.

All drawings were created using EASYL™ with Deluxe Paint™ on the Amiga computer and output on an Ink-Jet printer.



# Forth!

## A Tutorial

By Jon Bryan

This month we begin the process of designing a real application for the Amiga. "Begin" is the best way to describe it, as I will be writing it as we go along. I will be learning the machine right along with the rest of you, and perhaps I can communicate some of my problems (and hopefully solutions) as the program evolves.

What I have in mind is a three-dimensional simulation of a bouncing ball, complete with gravity, friction, and sound. It may reinforce the Amiga's image as a game machine, but it will be a lot more fun than writing a terminal program or an extension to AmigaDOS. Some aspects of the program will no doubt be primitive, and others less than optimum, but it should serve as a jumping-off point for other, more elaborate applications.

### IT'S ALL BEEN DONE BEFORE

The program won't be written completely from scratch, since I have already written a version for the Commodore 64. Most of the work will involve translating the graphics and sound specifics from one machine to the other. I wrote the version for the 64 using Parsec Research's SUPER FORTH 64, and judging from the level of performance which I was able to obtain, we should be able to achieve some very impressive results on the Amiga.

I expect to use a mixture of top-down design and bottom-up coding on this application. We'll start the design phase this month, and throw in some code as the project progresses just to keep everyone interested and to give you something to play with. My rough estimate at this point is that the project will require about four installments, but that could change depending on your feedback and the level of sophistication of the final product. There will have to be some design trade-offs involved. I don't expect to produce the be-all and end-all of Forth programs for the Amiga. The primary constraint is one of time: I have a "real" job to which I have to devote myself. Writing this column is something I do on the side. The other constraint is space.

### PROCEEDING TOWARD A DESIGN

The basic idea behind the program is simple. I want to create a reasonably realistic simulation of a rubber ball bouncing inside a closed space. The screen of the monitor will provide a window into the space through which the action may be observed. It will probably look something like the view of a handball or racquetball court -- as if we were spectators viewing the action through the glass back wall.

The calculations involved can be reduced to a few simple equations. For the sake of realism, our primary concern will

be to generate a convincing perspective view. Figure 1 illustrates the problem. If you look at the picture, and remember some simple geometry, the rule of similar triangles requires that the following relationship be true:

$$x/D = X/(D+Z)$$

Solving for x we arrive at

$$x = X * D / (D + Z)$$

This equation gives us the means to project the ball's position (in one axis) onto the "window" of the CRT. This basic equation will also be used to determine the proper size for the ball at varying distances. I'll be going into greater detail as we progress toward the solution.

Moving along rather quickly, the other aspect of the problem involves calculating the ball's motion. We can't project the ball onto the screen if we can't predict where it's going to be. In this simple simulation the forces influencing the motion of the ball will be limited to gravity and friction.

The effect of gravity on a falling object can be described by two simple equations (refer to any high school physics text):

$$D = (V_0 * T) + (A/2) * T^2$$

where D is the distance traveled, V<sub>0</sub> is the initial velocity, T is the elapsed time, and A is the acceleration of gravity, and

$$V_1 = V_0 + (A * T)$$

where V<sub>1</sub> is the final velocity.

We can arbitrarily, and conveniently, say that one unit of time will pass each time we move the ball, since we'll be doing it in a loop. Then, when we make T=1 (and cancel units), the equations simplify to:

$$D = V_0 + (A/2)$$

and

$$V_1 = V_0 + A$$

Since the value for acceleration can be considered a constant, A/2 is a constant as well, thereby reducing the calculations required to handle the acceleration due to gravity to two additions.

There are two sources of frictional loss which will have to be taken into account, since in order to be realistic the ball's velocity should gradually decay. One aspect of that decay is expressed as the "Coefficient of Restitution." In simple



# OH, SAY CAN YOU 'C'!

When "Key to C" was first introduced, AMIGA microcomputer programmers responded enthusiastically. Now, there's a new, extensively enhanced, even better version! The 'C' functions are similar to BASIC. The object library's good, clean working code includes windows, screens, menus, graphics, requestors, and alerts. For even greater productivity, we include our own system utilities.

## UNLOCK THE MYSTERY WITH THE KEY TO 'C'

- Source & Executable Code • Faster & Easier
- Full Documentation • Deliveries Begin Sept. 1

\$34.95



**DATA RESEARCH PROCESSING, INC.**

5121 Audrey Dr.  
Huntington Beach, CA 92649  
Phone: (714) 840-7186

## End Guru Meditation Errors

### INTRODUCING AMIGA DOS & MICROSOFT® BASIC TEMPLATES



- Quick reference to all commands
- Fits over keyboard — made of durable vinyl
- Professionally designed

Get the Guru Busters!™

☐ Amiga DOS \$9.95 ☐ Microsoft Basic \$9.95 ☐ Both \$16.95

Charge my: ☐ Visa ☐ Master Card ☐ Check/money order

Credit Card No. \_\_\_\_\_ Exp. Date \_\_\_\_\_

Name \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ ZIP \_\_\_\_\_

Signature \_\_\_\_\_

Michigan residents add 4% sales tax. Add \$1.50 shipping & handling

Mail to: Slipped Disk • 31044 John R • Madison Heights, MI 48071

Dealer inquiries welcome. Phone: (313) 583-9803

Allow 2-4 weeks for delivery

terms, no ball is perfectly elastic: it loses energy on each bounce. The easiest way to express the coefficient is as a percentage. A very springy ball might retain 90% of its velocity on a bounce while a less springy ball might have a coefficient of 50%. Whenever the ball strikes a surface we'll have to scale its velocity according to its coefficient of restitution, which will be a variable (and easy to change).

There is another small problem involving friction to be solved. The ball also loses energy to friction when it's rolling. If we don't factor in this source of friction the ball will only stop when it "sticks" to one of the walls, since it will only lose energy on a bounce. Once the ball has struck a surface it's a simple matter to change the sign of the velocity component perpendicular to that surface, causing the ball to "bounce." Detecting contact with a surface is straightforward as well. There are a couple of other problems which aren't so simple, but I'm going to leave you to ponder the implications, and cover them in a later installment.

### SO WHEN DO WE ACTUALLY GET TO FORTH?

The Forth program to simulate a 3-D bouncing ball could be expressed like this:

```
: Bounce-the-ball (--)
  Initialize-the-odds-and-ends
  BEGIN
    Move-the-ball
    Bounce-if-it's-necessary
    Has-it-stopped?
  UNTIL ;
```

which is very wordy, but would be perfectly acceptable. I actually plan something a bit more complex, perhaps like this:

```
: Bounce-it (--)
  Initialize
  BEGIN New-velocities
    BEGIN Move Bounce Stopped? UNTIL
  AGAIN ;
```

which would bounce the ball endlessly, giving it a new set of velocities each time it came to rest.

I'm going to quickly define a few of the words contained in "Bounce-it", then sign off for this installment. Please remember that everything you see here is subject to radical change.

```
: Move (X position\Y position -- New X\New Y)
  New-Z New-X New-Y Clip Move-ball ;
```

I've departed from standard practice by labeling the horizontal and vertical axes as X and Y respectively. A physics textbook would probably label the vertical axis Z, but I've chosen to use it to represent depth. My reasoning is that everyone is already used to calling the screen axes X and Y. You're probably thinking to yourself that "New-Z" came out of nowhere. I'm hiding the details until later, since in the earlier version of this program I used a variable for the Z-axis position to avoid what I thought was an excessive



amount of stack manipulation.

```
: Bounce ( X position \ Y position )  
Front/back Sides Top/bottom ;
```

"Bounce" checks to see if a collision has occurred and handles it accordingly. For instance,

```
: Top/bottom ( Y position )  
DUP Ymin = OVER Ymax = OR  
IF Yvel Enough?  
IF Blip THEN  
Yvel Reflect  
THEN ;
```

I know there are a number of things left implied here, but I want you to have something to think about. I'll only include one more word.

```
: Stopped? ( Y position -- Y position \ Flag )  
DUP Ymax -  
Xvel @ OR Yvel @ OR Zvel @ OR  
NOT ;
```

That should give you some food for thought. Keep in mind that the ball has to be ON THE FLOOR to be "stopped," and that on the screen the Y axis numbering starts from the upper left corner.

#### OTHER ARBITRARY DECISIONS

I will be writing the program using Multi-Forth from Creative Solutions. One immediate result of that decision will probably be that the Z-axis position value will be kept on the stack. Multi-Forth has "local" variables, so the stack manipulation problem will go away. There will be some tips on porting it to other implementations of Forth for the Amiga, but my main concern is to make something work and communicate the details to you. I think excessive generality would confuse more than enlighten, so those of you who have other implementations of Forth will have to bear with me, and do some translation on your own. I will try to help ease some of the burden.

There are two ways to approach the solution to this problem. One way is to get the job done with the least amount of fuss. The other way is to include such things as a shaded ball, a shadow and a dynamically sizeable room. While we are at it we could include sliders to vary the ball's elasticity, the force of gravity, and the friction coefficients. There could be multiple balls (and collisions between them). The possibilities are endless, and I expect to come down somewhere in the middle. I'd at least like to have a shaded ball which looks like a sphere, and I'd also like to throw in a shadow to improve the illusion. However the final version turns out, it should provide a broad-based introduction to the many features of the Amiga.

•AC•

## A GREAT COMPUTER... OUTSTANDING SOFTWARE... ...AND INCREDIBLE PRICES!

There may not be a better personal computer than the Commodore Amiga. But no computer can be better than the software that runs on it. Micro-Systems Software, makers of OnLine! and BBS-PC for the Amiga, proudly announce another link in their chain of value-packed software.

Analyze! is a powerful electronic spreadsheet program. Essentially, this program is a full-screen calculator where you can organize your data into rows and columns. These rows and columns can be analyzed with simple mathematics or complicated formulas. Rows and columns can be duplicated to avoid re-typing. Both data and formulas can be edited with only a few keystrokes.

From home budgets and check registers to financial modeling and your company's general ledger, all manner of bookkeeping tasks become faster and easier with Analyze! An outstanding value at only \$99!

OnLine! combines features and convenience in a high quality package that will meet all your telecommunications needs. With OnLine!, you can use your Amiga as a window to the world of information that is just on the other side of your telephone. You can link up with commercial information services for stock quotes, airline information and reservations, technical databases, and thousands of other business and entertainment tasks. You can also plug into local bulletin board systems (BBS) and discover a new world of information and software for your computer. Corporate users can use OnLine! to let their Amiga access data stored on the company's mainframe computer.

OnLine! is the finest program of its type available for the Commodore Amiga. You can't lose when you get "online" with OnLine!. All for a down to earth price of only \$69!

2400 bps modems! 2400 bps modems are breaking the speed barrier in telecommunications, and Micro-Systems is breaking the price barrier in 2400 bps modems. Transfer files 2 times faster than a 1200 bps modem and 8 times faster than a 300 bps modem. Micro-Systems will sell you a Hayes Smartmodem compatible 2400 bps modem, a special Amiga serial cable, and a copy of OnLine!, all for \$429.

That's right, the modem, the cable, and the software, all you need to begin using your Amiga as a terminal to the world, priced at \$429! Hundreds less than our competition!

### Micro-Systems Software, Inc.

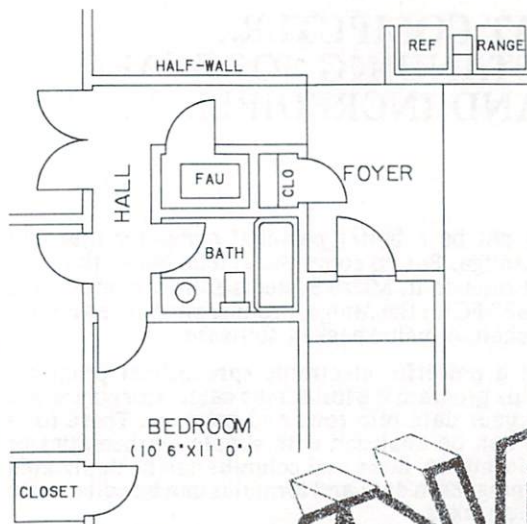
4301-18 Oak Circle  
Boca Raton, FL 33431

(800) 327-8724/National, (305) 391-5077/Florida

Ask us about our Amiga bulletin board program, BBS-PC.  
The first BBS for Amiga!

AMIGA, OnLine!, Analyze!, BBS-PC, and Smartmodem are trademarks of Commodore-Amiga, Inc., Micro-Systems Software, Inc., and Hayes Microcomputer Products, Inc., respectively.





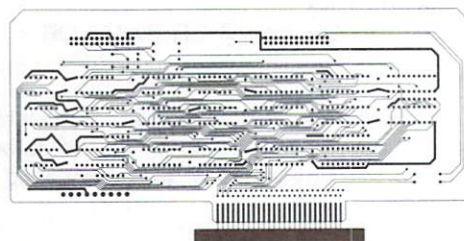
Finally,  
a proven  
CAD system for the Amiga

# DYNAMIC-CAD

FROM MICRO-ILLUSIONS

What AutoCAD\* can do for the IBM\*, Dynamic CAD can do with the Amiga... and a great deal more for less than a fourth of the price!

PRINTED CIRCUIT ARTWORK



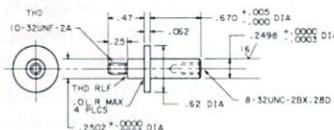
Emerging from years of successful problem solving applications in piping, and electronics for the aerospace industry, DYNAMIC CAD has brought a highly advanced and powerful CAD system together with today's most dynamic and versatile micro-computer, the Amiga. DYNAMIC CAD takes full advantage of Amiga's extensive capabilities with color, multiple modes of resolution, mouse functions, and easily accessible pull-down menus.

This is not some promised "vapor-ware." DYNAMIC CAD exists now and comes to the Amiga with a proven track record. The time and money-saving applications of DYNAMIC CAD for engineers and architects are truly astounding. Here is an advanced, 2-D drafting system with isometric capabilities that can be combined with many models of printers, plotters, and digitizers. In getting started you'll have the support of an extensive manual written in understandable English along with working examples as tutorial lessons.

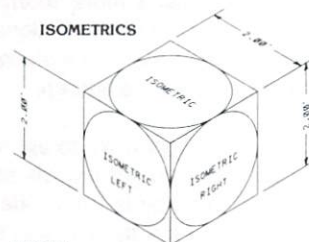
## WHAT DYNAMIC-CAD CAN DO FOR YOU

- D-C gives you all the expected CAD functions of zooming, rotating, panning, group functions and menu driven features.
- D-C brings you professional CAD capability tested and proven in the production of tens of thousands of drawings.
- D-C will liberate you from the need to draw free hand.
- D-C has net listing capability from your schematic.
- Schematic comparison to your printed circuit artwork for continuity check.
- D-C can produce isometric views.
- Mil-Spec quality Leroy\* fonts.
- Automatic line dimensioning.
- D-C includes a series of information libraries: Symbols, Electronic Parts/Chips, Architectural Components, Landscaping, etc.

- Data base to store and retrieve information on parts specifications, vendors, and pricing.
- Data base system utilizes ASCII format files which are convertible to other standards.
- Capable of utilizing up to 4,096 colors.
- D-C can generate over 8,000 layers.
- D-C supports most standard dot matrix printers, ink jet, laser jet, pen plotters, and the Gerber\* Photoplotter.

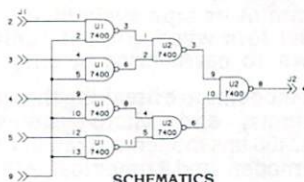


MECHANICAL DRAFTING



SYSTEM  
REQUIREMENTS  
512 K RAM  
2 Disk Drives (or)  
1 Drive and Hard Disk  
Printer or Plotter

Inquiries invited. (818) 360-3715

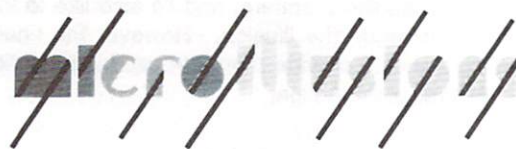


SCHEMATICS

| SIGNAL NAME | COMP NAME | FIN. NUMP | SOURCE | TYPE |
|-------------|-----------|-----------|--------|------|
| S0001       | U1        | 2         | 1      | CDN  |
| S0001       | U1        | 1         | 1      | 7400 |
| S0002       | U1        | 3         | 1      | CDN  |
| S0002       | U1        | 4         | 1      | 7400 |
| S0003       | U1        | 4         | 1      | CDN  |
| S0003       | U1        | 9         | 1      | 7400 |
| S0004       | U1        | 5         | 1      | CDN  |
| S0004       | U1        | 12        | 1      | 7400 |
| S0005       | U1        | 9         | 1      | CDN  |
| S0005       | U1        | 2         | 1      | 7400 |
| S0005       | U1        | 5         | 1      | 7400 |
| S0005       | U1        | 10        | 1      | 7400 |
| S0005       | U1        | 15        | 1      | 7400 |
| S0006       | U2        | 3         | 1      | 7400 |
| S0006       | U2        | 1         | 1      | 7400 |
| S0007       | U1        | 6         | 1      | 7400 |
| S0007       | U2        | 2         | 1      | 7400 |
| S0008       | U1        | 8         | 1      | 7400 |
| S0008       | U2        | 4         | 1      | 7400 |
| S0009       | U1        | 11        | 1      | 7400 |
| S0009       | U2        | 5         | 1      | 7400 |
| S0010       | U2        | 9         | 1      | 7400 |
| S0011       | U2        | 6         | 1      | 7400 |
| S0011       | U2        | 10        | 1      | 7400 |
| S0012       | U2        | 8         | 1      | 7400 |
| S0012       | U2        | 4         | 1      | CDN  |

NET LIST FROM SCHEMATIC

\*Gerber — trade mark of Gerber Scientific Instruments  
\*Leroy — trade mark of Keuffel & Esser  
\*IBM — trade mark of International Business Machines  
\*AutoCAD — trade mark of Autodesk, Inc.  
\*Amiga — trade mark of Commodore Business Machines



P.O. BOX 3475, GRANADA HILLS, CA 91344



# Roomers

## By The Amigo

OK, about that new hardware I promised you last issue. At Comdex, Commodore unveiled several prototypes for their future Amiga. No, this isn't Ranger ... remember what I said about the Ranger being an ideology ... open ended, expandable, etc.

Instead, the newly designed Amiga could be called 'the business Amiga.' According to the prototypes displayed, it will be your basic Amiga inside, except the 68000 CPU will be replaced by a 68010, giving it about a 10% general increase in speed. The machine is commonly referred to as the A2000, but I have also heard from reliable sources that it is going to be called the A2500.

The reason I call it 'the business Amiga' is the Sidecar (see last month's Amazing Computing for more info on the Sidecar) will be built in. That means native AmigaDOS and MS-DOS both built-in. Each side (68010 and 8088) will run at full speed, simultaneously, and can share resources, just as the Amiga A1000 and the Sidecar currently do. The new machine may have two built-in drives: a 3-1/2 inch drive and a 5-1/4 inch drive. The new machine's case is metal, instead of the A1000's plastic case, and the keyboard has a new layout - among other things, the function keys come down the left side of the keyboard rather than across the top. Lastly, there may be up to six (I keep getting differing reports, six seems to be the most consistent number) Zorro expansion slots in the machine.

### Also coming from Amiga

A new chip set that can address 2 MB of memory, instead of the 512K that they address now. If you look at the Amiga's memory map, you'll see a space of 1.5 MB that's "reserved for future use." Well, now you know what that future use is. It's expanded chip memory. That means (for example) that many images can be stored in chip memory and quickly cycled through for animation effects, or sound tables and song data can be long enough to store data for a half an hour of music. Lastly, Amiga has shown some of its reps a DMA SCSI disk drive, with the warning that "we are not committed to this as a product." So we may see it, we may not.

**Little known (until now):** Current Amigas are shipping with a new keyboard. This keyboard does not have the "parallelogram problem" that the old keyboard has. Try this test: Hold down the A, Z and X keys at the same time. If you also get an 'S', you have an old keyboard (join the club).

**Some trivia:** Irving Gould, head honcho at Commodore (and holder of 6 million shares of stock) looked at the Emulator and didn't like what he saw. So he hired some crack engineers from Germany to develop the Sidecar.

In applications, CBM is trying to convince the "big three" auto makers in Detroit that they should standardize on the Amiga for inventory and parts ordering. Not only would that mean on the factory floor, but all the companies' distributors and dealers would have Amigas too. If it goes through and some swift software people realize what's good for them, they stand to make a ton of money from special applications like this. Well? What are you waiting for?

Apparently Commodore wasn't too sure what it should do about prices of the Amiga, so it quietly extended its 'buy an Amiga and get a monitor' deal for another month. The latest deal, starting July 1, is that the price of the Amiga is dropped to \$1095 (from \$1295) and the monitor drops from \$495 to \$395. Now if you buy either by itself, you get a better deal, but buying both will really cost you \$1490, which is \$195 more than the old 'package' deal. The reason for this is that CBM was losing its shirt with the package deal, and this should even things out a bit. But if you take a look at the CBM warehouses, you'll see a huge stockpile of 5-1/4" drives and A1080 monitors ... they just can't seem to get rid of them.

Speaking of deals ... a dealer up in New England tells me that they've started what he calls a 'bird-dog' program. If you buy your machine at his place and then drag in a friend to buy an Amiga, you get 50 bucks. Some dealers are giving cash, some are giving merchandise credit. This dealer was under the impression that it was a New England only program, but not every dealer was going for it. If you're in that area, ask around.

Many people have been yanking out their 68000 CPU and replacing it with a 68010 CPU. This is an inexpensive, operation (although it voids your warranty) and provides a performance boost of about 5%. You can get up to 20% improvement in math-intensive applications, too.

Instructions on how to do this are on Fred Fish disk #18 (available from PiM Publications). Now that Motorola has announced a 50% price cut in their 68020 and 68881, we may start to see more people opting to build "Turbo Amigas", which will dramatically boost performance.

The hardware types at Amiga looked at the 'final' prototype of the Genlock, and after much scrutiny, decided the resulting picture quality wasn't good enough, so they've redesigned it. Now Commodore is talking about October as a release date. Meanwhile, those folks at A-squared have their LIVE! board jumping through hoops. They are generating 10 HAM images EVERY second! That's almost LIVE! No word on whether the software that accomplishes this will be available with the machine; let's hope so!





## YET ANOTHER UNFAIR ADVANTAGE.

Although you haven't had your Amiga for very long, you may find that you need a more powerful line interpreter. Consider these features:

- Pipes
- Search paths
- User definable command-line editing
- Definable function keys
- Unix-like wildcards
- More versatile redirections
- Command aliases
- Built in commands
- Command history

All available now, at a reasonable price, from

# Z O X S O

THE AMIGA TOOLSMITHS.  
P.O. Box 283, Lowell MA, 01853 0283 USA

Unix is a trademark of AT&T  
Amiga is a trademark of Commodore Amiga Inc.

Speaking of Genlock, a company in the Washington DC area took a look at the Genlock from Amiga and said "Yuk!" They are producing their own professional-quality unit that will sell for about \$1500. Word has it that there are 21 BNC connectors on the back of this thing!

Latest word on release 1.2 of the system software is that it will be available sometime in August. I doubt it; look for September as a release date. However, when it does come out, you won't be disappointed. Faster directories, faster icons, less crashing, CLI search paths, RAM-resident commands that you can enable, Preferences support for the Serial device, new printer support, faster RAM disk, RAM: icon from the workbench, new disk tools, new graphics functions, auto configuration & expansion architecture support, PAL support, an expanded notepad utility and more make this an exciting release. Another feature of V1.2 is that the DOS supports partitioned disks; you could have OS-9 on part of your hard disk, AmigaDOS on another part, and MS-DOS on yet another part! Whew!

The toolbox disk that will be forthcoming from Amiga will be helpful too. The FontEditor looks great! I still don't know what we'll see for delivery date or price, though.

Speaking of fonts, be on the lookout for new fonts! A small company is about to release over 100 fonts for the Amiga that can be used in notepad, paint programs or your own custom programs! Last I heard, the individuals responsible were still deciding on a marketing strategy.

There's a new version of Scribble! out. It is available as a free update, does mail merge, spelling check and some other goodies. Also, a new version of Analyze! is out; it's an update that costs you money. Features include color, among other things. A new product, called Organize! is out. It is a database package. Look for a database package from a company called The Other Guys. Strange but true.

Are you looking for a color dot-matrix printer? Back in May at the NCGA (National Computer Graphics Association) show, Epson demonstrated a new printer that will sell for around \$500. People who saw the unit said the quality was great. I guess this is the printer that will take the place of the discontinued JX-80. Canon is also reported to have a new low cost (\$400) printer out soon.

If you are wondering why you can't get 300 dots per inch out of your HP Laserjet, it's because the driver was compiled at 100 dpi in draft mode and 150 dpi in letter quality mode. To remedy this, change the number stored at location \$0F2E (draft mode) and the number stored at location \$0F3C (draft mode) in the driver file to one of these values:

- 0100: 75 dpi
- 0200: 100 dpi
- 0300: 150 dpi
- 0400: 300 dpi

Speaking of laser printers, word is that the TeX text formatting package is done. Look for it soon.

If you haven't seen InfoMinder from Byte By Byte, go see it! It's gotta have the fastest disk drive access I've ever seen on the Amiga. A real classy product. \$90.

Look for ABSOFT (publishers of Fortran for the Amiga) to come out with a Basic compiler this month; price under \$300. The new version of Manx' Aztec C is out, which is compatible with release 1.2 of the Amiga system software, fixes some bugs, improves on its debugger, provides 68881 math coprocessor support, can generate output amenable to the Amiga assembler and supports scatter loading of code.

Roger Powell, formerly of now-defunct Cherry Lane Software, has emerged as Magnetic Music. So we will see Texture after all. However, Texture will only work with the Roland MPU-401 MIDI interface unit, and the MPU-401 doesn't work on the Amiga. Magnetic Music has the answer, however: an interface that allows you to hook the MPU-401 to your Amiga. Look for both products in August.

Look for MicroSmiths to announce their Mean 18 golf game for the Amiga. Also in the works: a package tentatively called Deluxe TxEd that will compete with such feature-packed packages as Microsoft Word. Hooray for MicroSmiths!

Mimetics said they will not be pushing sales of their MIDI interface; they simply built it because nobody else had one on the market. Now that other vendors are appearing with more capable devices, we may see the Mimetics interface just wither away slowly... Golden Hawk



Technology reportedly has developed a MIDI interface that has all the parts inside the connectors; however it is not as versatile as MIDI Gold, and no word if it will become a product.

You may have heard of Hippopotamus, who announced support of the Amiga. Recently they announced that their Amiga products have been put on hold "indefinitely," so don't hold your breath for them!

Metacomco has a toolbox disk coming out. Included will be support to use the serial device as a terminal, an enhanced CLI, pipes, and more. No word on release date or cost.

**Also from Metacomco:** APL for your Amiga! Really, they are simply doing the marketing; they did not develop it.

**Yet Another Terminal Package:** this one's called TeleCraft, and it's Amiga's entry. Its two claims to fame are that it supports 132-columns and the screen updates are fast; 9600 baud feels like 9600 baud, not 1200. I have not seen it, so I can't verify the throughput claim.

A company in Connecticut has developed a Commodore-64 Emulator. It was in beta test in June, so this thing is for real. It uses the 1541 or 1571 drive and runs full speed (I know, that doesn't say much for us speed demons). It is a hardware/software hybrid ... now your Amiga is three machines ... look for it soon!

Look for Alegria, a memory expansion unit from Access Associates. The board has 512K on it for \$379. But wait. If you want to buy higher density memory chips and install them yourself, you can buy their controller chips for \$10. With the denser RAM chips, you'll then have 2MB for about \$700. The expansion board does not pass the Amiga's expansion bus through.

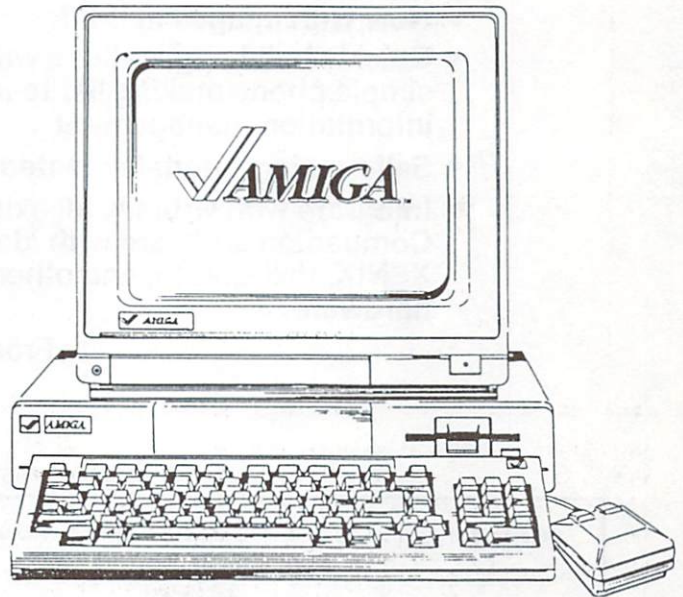
MicroBotics has announced their StarBoard II and its multifunction module. The StarBoard II is a 512K to 2MB RAM box. The module allows you to write protect memory, so if you get a GURU during development, you can flip the switch and write protect the RAM disk, then do your warm reboot, without losing the contents of the RAM disk. It passes the Amiga bus, and should be available early August. Also in the works from MicroBotics: a hard disk that plugs into the parallel port and has an SCSI port, and a DMA SCSI interface in the fall.

Word around is that AST has an Amiga and is excited about its potential; it looks good that we'll see some add-on boards from them sometime soon.

Another small company has an interface that lets you control 8mm VCRs and some standard VCRs through software. The main applications are video and digital sound; you could store over 24 hours of audio on a tape. It is VCR-specific; I'll let you know more as I know more.

•AC•

# ✓AMIGA HOUSE



(713) 988-3018

11600 S.W. FRWY., STE. B-216  
HOUSTON, TX. 77031

## HARDWARE

512 K Color System      Sony RGB Monitor  
Canon PJ-1080A Color Printer      Avatex 1200 Modem

## SOFTWARE

|               |                  |
|---------------|------------------|
| Activision    | Aegis            |
| Byte By Byte  | Chang Labs       |
| Digi-View     | Electronic Arts  |
| Infocom       | JHM              |
| Lattice       | Manx             |
| Micro-Systems | Mimetics         |
| Mindscape     | New Horizons     |
| TDI           | VIP Technologies |

Above is only a sample listing



# PUT YOUR AMIGA TO WORK

with

# DATAMAT™

FULLY RELATIONAL DATABASE MANAGEMENT SYSTEM.

- Now with images in IFF format, display with text/data/voice
- Quickly build applications without any program coding from simple phone/mailling list to research to organization-wide information management
- Self-running tutorials created automatically for personnel training
- Integrate with virtually all existing hardware systems  
Companion software with identical user-interface for MS DOS, XENIX, UNIX, VMS, and others available. Same application fits all hardware

From \$125.00

## DATAMAT PARTIAL SPECIFICATIONS

|                                                             |                                                                                |                                    |                                                                                                                        |
|-------------------------------------------------------------|--------------------------------------------------------------------------------|------------------------------------|------------------------------------------------------------------------------------------------------------------------|
| Organization                                                | Fully Menu-driven Relational Database Management System/Application Generator. | Number of data files per data base | Unlimited                                                                                                              |
| Number of characters per field                              | 1,024                                                                          | Data types                         | 13 includes Image in IFF Format                                                                                        |
| Number of fields per record                                 | 2,000                                                                          | Global (System) Fields             | 40 user definable<br>9 special purpose                                                                                 |
| Number of characters per record                             | 4,000                                                                          | Field checks                       | Mandatory, Type, Initial value, Value within a specified range.                                                        |
| Number of records per file                                  | 4.3 billion                                                                    | Password security                  | Field and data base levels                                                                                             |
| Multiple response                                           | Supports multiple responses (up to an array of nine) for a single field.       | Calculation capabilities           | Full complement of 23 math and trigonometric functions and 13 logical operators. Automatic date and time calculations. |
| Number of Relations per data file (simultaneous R/W access) | 10                                                                             |                                    |                                                                                                                        |

Data Entry - single entry to multiple files and records. Import/Export facility with data conversion/reorganization. Forms Definition - full screen editor with mini word processor. Report Generation - up to 66 lines x 132 columns, 6 level totaling with built in summary. Sort/Search - up to 26 selection criteria per query. Mass Editing, Time Saver Audit - stores all key strokes used in building application for automatic recreation. Statistics and Graphics- stepwise multiple regression, standard statistical tests and analysis; scatter plots, bar charts. Custom Applications Generator - batch/partial batch processing; user-defined menus; self-running demos.

Available through your Amiga dealers. Inquiries Welcome.



**Transtime  
Technologies  
Corporation**

797 Sheridan Drive, Tonawanda, New York 14150; Phone: (716) 874-2010

Datamat is a trademark of Transtime Technologies Corporation  
AMIGA is a trademark of Commodore Amiga, Incorporated  
MS-DOS & XENIX are trademarks of Microsoft Corporation

UNIX is a trademark of Bell Laboratories  
VMS is a trademark of Digital Equipment Corporation



# SCRIMPER

## SCReen IMage Printer

By Perry Kivolowitz

Usenet: 1hnp4!ptsfa!well!perry

This is the third and last installment of the description of Scrimper. In the previous two installments we've discussed how programs may cause screens to be printed by interfacing with the printer device, we've seen how menus can be constructed at run time and were introduced to a number of programming techniques and pointers.

This month, we'll discuss the interface between menus and Intuition as well as discuss a code saving technique in C, pointers to functions.

To wrap up some loose ends from last month let's look at how the list of screens is generated and turned into a menu. Turn to the routine, `build_menu`. This routine will call another routine `"get_screens"` to assemble a list of screens then for each screen in the list plug its name into a menu structure.

The first thing of note is that one of two parameters to `build_menu` is not used. The parameter `"scrn"` was left over from a previous version of `scrimper`. Leaving `scrn` in as a parameter is particularly wasteful in this case. Passing unneeded parameters is always a waste because code must be generated to place the parameter on the stack. In this case however, I had made `scrn` in to a register variable which means that each time I invoke `build_menu` I waste a whole c.p.u. register which might otherwise be used. The lesson here is two fold: First, don't pass parameters that won't be used. Second, register variables are to be chosen with careful consideration as to frequency and complexity of use.

The next statement to be executed, a call to `alloc_menu_item`, assumes that there will be at least one screen found by `get_screens`. This is a reasonable assumption since `scrimper` itself must be running from something, probably the workbench screen.

If `alloc_menu_item` fails it will return false and we'll in turn return false from `build_menu`. Next I initialize `p`, a pointer to `MenuItems`, to point to first item in the menu which the first time through the coming loop will be the item created to represent the first screen currently known to the system.

I go into a loop now, notice that the first value of the loop counter is one (not 0). This is because the zeroth iteration of the loop is "hard-coded" outside the loop. This is an example of a time honored programming technique which in this case wasn't really needed.

The technique is to "prime the pump" before entering a loop then "pumping away" by entering the loop. This

is usually used when the first iteration of a loop is substantially different than the following iterations. Usually this construct is used for reading data as in:

```
Set up first read which might need special code.
Do first read.
Do {
    Processing;
    Attempt Next Read;
} While (Read Went OK);
```

In `build_menu`, first call to `alloc_menu_item` and the succeeding "if" statement could be done away with by noticing that the needs of the first iteration are really the same as every other iteration. Notice that `FirstItem`, a field in a `Menu` structure is of the same type (a `MenuItem`) as the linkage field `NextItem` contained in each `MenuItem` structure.

Thus instead of:

```
register struct MenuItem *p;

if (get_screens()) {
    alloc_menu_item(&menu->FirstItem, 0, titles[0]);
    if (!menu->FirstItem) return((int) FALSE);
    p = menu->FirstItem;
    for (i = 1; i < screen_count; i++) {
        alloc_menu_item(&p->NextItem, i, titles[i]);
        if (!p->NextItem) return((int) FALSE);
        p = p->NextItem;
    }
    return((int) TRUE);
}
return(FALSE);
```

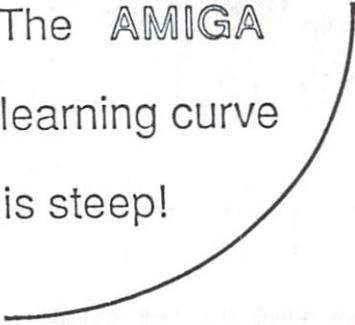
I could have written:

```
register struct MenuItem **p;

if (get_screens()) {
    p = &menu->FirstItem;
    for (i = 0; i < screen_count; i++) {
        alloc_menu_item(p, i, titles[i]);
        if (!*p) return((int) FALSE);
        p = &(*p)->NextItem;
    }
    return((int) TRUE);
}
return((int) FALSE);
```



The AMIGA  
learning curve  
is steep!



Choose a porting house that's  
well advanced along the curve!

**Advanced Systems Design Group**  
*Your Port of Entry into the Amiga Marketplace*

280 River Rd., Suite 54A  
Piscataway, N.J.  
08854  
1-201-271-4522

Amiga is a trademark of Commodore-Amiga, Inc.

What all this indirection means is the following: P is a pointer to a pointer to a MenuItem structure. That means p is the address of a place where the address of a MenuItem can be found.

For each screen, call `alloc_menu_item` which will place into the address specified by p, the address of an MenuItem (thus is p a pointer to a pointer). If the address stuffed into p (that is, \*p) is NULL then something went wrong in `alloc_menu_item`.

If nothing went wrong update to point to the space reserved for the pointer to the next MenuItem in the MenuItem just declared. Making this modification not only produces cleaner code but reduces the code size by a whopping 24 bytes (is Aztec 3.20A)!

Let's turn our attention now to the function `get_screens` which is called by `build_menu` to assemble a list of screen titles for use in the screen selection menu. `Get_screens` takes no parameters but leaves the global variable `screen_count` containing an accurate count of the number of screens. Or is that really so?

Can anyone guess at this point where the major (well not really so major) flaw is in the routine `get_screens`? I'll give you a hint. The second line of code establishes Scrimper's idea of what the first screen is. Notice that screen zero is taken from Scrimper's own window screen pointer. This seems safe right?

After all the Amiga EXEC stores the list of screens as a linked list of screens sorted by depth. The first screen on the list is the frontmost screen. The last screen in the list is the rearmost. Since Scrimper is responding to a user initiated event when it goes to build the screen list, Scrimper's window must be available to the user. Scrimper's screen must be the front most?

Right! But.....only some of the time.

If in fact the workbench screen is the front most screen when `get_screens` is called, Scrimper, as coded in the example code will accurately locate each currently defined screen.

But, take for instance the possibility that another screen is really the front most but has been partially dragged down to reveal the Workbench screen where the Scrimper window can be diddled. In this case, Scrimper will not catch the screens in front of the Workbench screen since they will precede the Workbench screen in the (depth) ordered list of screens.

How to correct this deficiency is left to the reader however, one hint might be "look at the first header file included by third header file included by Scrimper."

A part from this deficiency (which by the way "was" an oversight on my part where other "flaws" mentioned in this article were included for the purposes of discussion) `get_screens` is pretty uninteresting save only for the use of `Forbid` and `Permit`.

These two routines enable and disable multitasking on the Amiga. They are to be used extremely cautiously as misusing them will lead your machine down a dark and lonely path. I `Forbid` upon entering `get_screens` because the list of screens is a data structure which is shared by all processes using `Intuition` to access the Amiga display device.

As such I have to take steps to ensure that the data structure will not be modified while I am looking at it. To do this I shut down multi-tasking which means that while I am in the `Forbidden` state, my process will not be removed from the c.p.u. unless I explicitly volunteer to do so by calling a routine such as `Wait`.

It is critical that a routine which `Forbids` the processor must also perform a `Permit` to allow other tasks to run again. Thus it can truthfully be said that failing to call `Permit` is `Forbidden`! Notice, in `get_screens` there are two possible exits from the routine and each has a `Permit` immediately before it.

Now let's dive into the `getmenu` function. The neat thing about the `getmenu` function is that it manages two distinctly different menu structures performing completely different tasks yet does so with one set of code.

If we were to draw a picture of how an alternative means for accomplishing this it might look like so:



screen selection menu code handler -----> screen selection menu

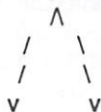


function selection menu code handler -----> function selection menu

That is, we have a section of code which manages the screen selection menu which for any selection executes the appropriate subroutines. One of the subroutines executed would be a routine to manage the function selection menu. There would be two routines in which the program would loop handling menu oriented messages.

The approach taken in scrimper would look like the following:

generalized menu code handler



screen selection menu    function selection menu

The way we do this is to abstract function calls which are menu dependent into a data structure which can be passed as a parameter to recursive calls to the generalized menu handling code.

The structure I defined is called "jmpTbl" for "jump table". A copy of the structure definition follows:

```
struct jmpTbl {
    struct Menu *menu;
    int (*init)();
    int (*cleanup)();
    int (*pick)();
}
```

This contains what may be a new construct for many readers. That is, pointers to functions. Pointers to functions are perhaps one of the most powerful constructs in the C programming language because they permit the design of algorithms with a very high degree of abstraction. Individual cases need not be coded if general purpose code can be written once and reused.

Let's look at pointers to functions are declared by comparing the declaration to others.

```
int A;
int *B;
int C();
int *D();
int (*E)();
```

A, of course, is the declaration of an integer. Memory large enough to hold an integer is reserved and the symbol A is taken to mean the contents of that location.

B is a declaration of a pointer to an integer. Memory is reserved for a location which will contain a pointer to another location which will contain an integer. Memory for the pointed

# DATA REDUCTION ASSOCIATES INC.

303 West Sixth Avenue Dept. A  
Tallahassee, Florida 32303  
904-681-0553

INTRODUCING A

## 68010 UPGRADE KIT...

INCLUDING;  
AN MC68010L8 CPU,  
"HANDLER()!"... AN EXCEPTION  
HANDLER THAT DOES NOT  
MODIFY YOUR CODE i.e.  
MOVE.W SR, <EA>, "DBG010"...  
AN INTERACTIVE DEBUGGER THAT YOU  
CONTROL, AND COMPLETE INSTRUCTIONS.

PRICED @ \$115.00 COD or CHECK.  
PRICE INCLUDES SHIPPING.

to integer is NOT reserved. B is taken to be the contents of a memory location which should contain the address of an integer variable. \*B is taken to be the contents of the integer variable.

C is a declaration of an integer function. C by itself is a pointer to an integer function but the pointer is fixed pointing to the first instruction of the "C" subroutine and cannot be reassigned. The value that the C function returns is a directly usable integer.

D is a declaration of a function returning a pointer to an integer D by itself is a fixed non-changable pointer to a function returning a pointer to an integer. D points to the first instruction of some subroutine somewhere.

E is declaration of a pointer to a function which returns an integer. E does NOT declare a subroutine, it declares only a pointer to a routine which must be defined elsewhere. E by itself is the value of the pointer in the same way that B by itself is the value of a pointer. In the same way that \*B is the thing pointed to be B, (\*E)() is a CALL to the function pointed to be E.

Note that E is different from D or C in that E it is okay to change the value of E while the value of D and C are fixed. The following statement assigns makes E point to the same place as C and in effect calls the C subroutine twice.

```
E = C;
C();
(*E)();
```



Why would we want such a construct? Well, it allows us to compose abstract algorithms without consideration of specific cases. For example, we were writing a sorting algorithm, the sorting algorithm is exactly the same if what we sort are integers or strings, right? In some ordering some integers come before others. Similarly in some ordering some strings come before others. The sorting algorithm itself remains the same.

When we try to translate the sorting algorithm into a programming language we find that we'll have to have one routine which sorts integers and another that sorts strings. But wait! Why not abstract from the sorting code the statements which do the comparison of one item to another? In this way, the bulk of the code and in essence the spirit of the algorithm will not need duplication for each variable type we wish to sort.

Instead of having a sorting routine for integers and another for strings we can have exactly one sorting routine and pass it a pointer to a routine coded to compare two variables of a specific type.

For example the following routines might be used for comparing variables to integer and floating types:

```
compare_integer(a, b)
{
  if (a > b) return(1);
  else if (a < b) return(-1);
  return(0);
}
```

```
compare_floating(a, b)
float a, b;
{
  if (a > b) return(1);
  else if (a < b) return(-1);
  return(0);
}
```

In *scrimper* I use the *jmptbl* structure to point to routines which will perform menu strip initialization, the execution of menu selection specific code, and menu strip clean up for varying menus (in this case, the screen selection menu and the function selection menu).

```
The general flow will be:
while (no errors and no close gadget) {
  call getmenu with screen selection menu
    handle menu selection
    call getmenu with function selection
      handle menu selection
      return from getmenu
    return from getmenu
}
```

Notice the recursive call to *getmenu*. This strategy may not be a big winner (in terms of code size) for just two menus, but suppose there had been a dozen inter-related menus and sub-menus? This sort of recursive design would save

considerable amounts of code as well as provide a clearer impression of what the inter-relationships are.

We call *getmenu* with a pointer to a *jmptbl* structure. The first thing *getmenu* does is invoke the menu strip initialization routine pointed to by the field *init* in the *jmptbl* structure.

In the case of a screen selection menu, the *init* field point to the routine, *screen\_init*. *Screen\_init* builds a fresh list of what screens are currently defined to the system by calling *build\_menu* which has been described previously.

If *build\_menu* returns an error then *screen\_init* will return an error. If there is no error the newly constructed menu is prettied up a bit by calling *message\_left\_edges* which centers each menu item in the space occupied by the widest item in the menu.

Next we call *SetMenuStrip*, an *Intuition* routine which takes a pointer to a window as well as a pointer to a *Menu* structure. *SetMenuStrip* tells *Intuition* that it should prepare to render menus according to the specification contained in the pointed to *Menu* structure.

Recall that a *Menu* structure defines where across the menu strip the heading of the menu will appear and how wide this particular menu's header selection box will be.

It is definitely not possible to activate a menu for a given window until *SetMenuStrip* is called. However, calling *SetMenuStrip* in and of itself does not complete the steps required in enabling the proper handling of a menu.

Other steps include having the IDCMP message *MENUPICK* enabled. Notice I set this flag when defining the contents of the *NewWindow* structure. This means that whenever menu events occur while *scrimper*'s window is active, notification of these events will be sent to *scrimper* via IDCMP messages.

Let's continue on a bit. After calling the menu initialization routine we enter a loop which will terminate only by exiting *getmenu* as a result of receiving an IDCMP message.

We attempt to get a message from the IDCMP (*w->UserPort*) by calling *GetMsg*. Recall from an earlier *Miga-Mania*, *GetMsg* will return a *NULL* pointer if no message is available. Hence the test on the following line.

If no messages are queued for *scrimper* I remove *scrimper* from c.p.u. contention by calling *Wait*. This has the effect of placing the *scrimper* process in a sort of suspended animation. While in such a state *scrimper* will not consume any c.p.u. time allowing the full computational resources of the Amiga to be used for other concurrently executing programs.

The ROM kernel will awaken *scrimper* if there is any activity on *scrimper*'s IDCMP (a clear case of "don't call us, we'll call you").

If we get past the if statement it means that message points to a valid *IntuiMessage* structure. Notice that the very first



things I do is to store away certain key fields of the IntuiMessage. Can you recall from the Miga-Mania in issue 3 of Amazing Computing why this is done?

It is because the space occupied by the IntuiMessage structure is being loaned to us by Intuition. As soon as we signal to Intuition that we are through borrowing the IntuiMessage (which should be as soon as possible) Intuition will reuse the location for some other purpose. That is, we cannot be assured of sole access to the borrowed memory starting at the point at which we call ReplyMsg.

So, our choices are either to not call ReplyMsg until we're done with the message or to save the parts of the message we'll need and ReplyMsg right away. Which way should we go?

Well, whenever you borrow somebody's prize possession (and main memory can be argued to be the most sought after possession in the Amiga) you should try to return it as soon as possible so we'll go the second route of saving the parts of the IntuiMessage we need and ReplyMsg'ing right away.

For scrimper's purposes we need only to save Class and Code. Class will tell me what kind of IntuiMessage I just received (this is the way I tell a CLOSEWINDOW strike from a MENU PICK). Code will tell me exactly which menu item was selected IF the Class was MENU PICK (the interpretation of Code is dependent upon the value of Class).

Code and Class are saved and then the message is replied to immediately. Now all we have to do is act upon the received message. To do this we switch upon the saved value of Class.

If the message was due to a CLOSEWINDOW message it means that the user has detonated the close gadget in scrimper's title bar. If this is so, we'll have getmenu return FALSE. Returning FALSE from getmenu means either that there was some error somewhere or the user wants to exit. In either case the FALSE value will be percolated up through any nested calls to getmenu finally causing the top level "while" loop in main to terminate.

If the message was due to the user selecting another screen or window Intuition will send scrimper an INACTIVEWINDOW message. This is needed to handle that gizmological "screen-to-front" feature of scrimper's.

If the user had depressed the LMB (left mouse button) while over scrimper's own window an ACTIVEWINDOW message would be received. Between ACTIVEWINDOW, INACTIVEWINDOW and CLOSEWINDOW we've provided a way of intercepting any left mouse button depression which might occur.

As a side, how might I save a few lines of source code from this switch statement? Like so:

```
case CLOSEWINDOW:
case ACTIVEWINDOW:
case INACTIVEWINDOW:
(*jmpTbl->cleanup());
```

## AMIGA 256K CARD

Only \$ 99.00

## 1 YEAR WARRANTY

AMIGA GIVES YOU A CREATIVE EDGE.



**MICHIGAN SOFTWARE  
DISTRIBUTORS INC.**

43345 GRAND RIVER • NOVI, MICHIGAN 48050

TELEPHONE (313) 348-4477

MODEM (313) 348-4479

```
return((int) !(class == CLOSEWINDOW));
```

This brings us to the MENU PICK messages where we must look at the saved value of Code to determine exactly which menu item had been selected.

The first thing to observe is that the previous sentence should read "exactly which menu ITEMS had been selected." That is, Intuition allows multiple menu items to be selected by depressing both mouse buttons while over one or more menu items.

When more than one menu item is selected during a single menu event Intuition defines a linked list of selected menu items within the MenuItem field, "NextSelect." Actually this isn't exactly so. NextSelect does not contain an absolute memory address specifying the next MenuItem structure to be examined but rather it contains a compound value representing menu number, item number and sub-item number if appropriate.

Menu number starts from 0 and counts upwards from left to right. Item number starts from 0 and counts upwards from top to bottom. Similarly for sub-item numbers where appropriate.

Macros exist for converting these compound symbols for MenuItems into a more useful form. For example, given a pointer to the head of a menu data structure and a code from a MENU PICK message, ItemAddress will return a pointer to the MenuItem specified by the code or a NULL pointer if no corresponding item exists.



CARDINAL ANNOUNCES ITS

# 2 x 2

## EXPANSION DISK DRIVE

GIVE YOUR AMIGA  
2.6 MEGABYTES

### 2 x 2 CONSISTS OF:

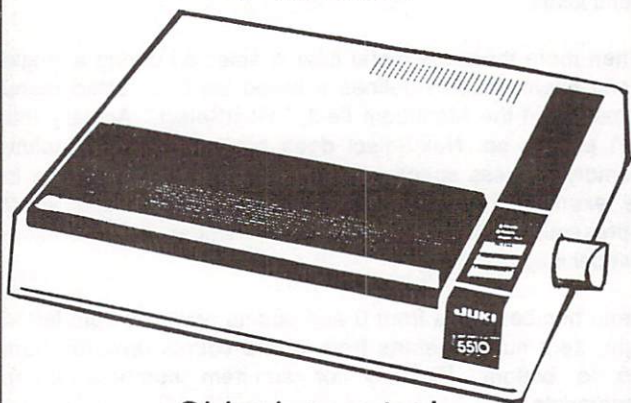
2 5 1/4" 80 track drives,  
electronics and software.

- AMIGA DOS MODE - Emulates 3.5" drives with 880K each.
- PC DOS MODE - Provides dual 40 track, 360K drives.

**\$595.00**

### JUKI 5510 C Color Printer (Uses Epson Codes)

**\$469.00**



*Shipping extra!*

Cardinal Software 14840 Build America Drive  
Woodbridge, VA 22191 Info: (703) 491-6494  
Order now! (10-4 Mon-Sat)



**800 762-5645**



I decided, and I'm sure you'll agree with me, that for scrimper's purposes multiple menu selection don't really make sense. So, I have to filter out any multiple selections if they had occurred by looping through what may potentially be a link list of menu items to find the valid member of the chain.

Once I'm satisfied that code contains the compound symbol for the last MenuItem selected I pass it to the routine pointed to by the "pick" routine followed by a call to the "cleanup" routine.

Notice I pass back the value returned by the pick routine as the value of this instantiation of getmenu (this is how the percolation mentioned before is accomplished).

If we had just made a screen selection the pick routine which would be called is screen\_pick. The first thing we do in screen\_pick is record which screen was selected by saving the item number extracted from "code." If the first MenuItem had been chosen the value of ITEMNUM(code) would be 0. If it had been the second MenuItem the value of ITEMNUM(code) would be 1 and so on.

Having recorded the user's screen selection let's find out what the user wants to do with the selected screen. To do this we call getmenu again! This time, we pass it the function selection jmptbl rather than the screen selection jmptbl.

This causes the menu strip containing the screen selection menu to be replaced with one containing a function selection menu. Screen\_pick returns the value returned to it by its own call to getmenu.

Calling getmenu with the function selection jmptbl will probably lead to calling func\_pick (should the user make any valid menu selection). Func\_pick attempts to verify that the selected screen is still valid (since time may pass between generating the list of screens, displaying it, and having the user select it plus a function to perform) by rescanning the list of screens looking for the one we're hoping for.

If the hoped for screen is found in the system's list of current screens (even if it had moved) verify\_screen\_pointer will return a pointer to the screen structure representing the screen selected. If the screen selected became undefined (the list of screens no longer refers to the hoped for screen) since its selection, verify\_screen\_pointer returns a NULL.

If the screen is a valid one we look at the saved value of Code to determine which function should be performed upon the screen. MenuItem 2 (the third menu item in the function selection menu) simply causes scrimper to return to the screen selection menu.

If the user has chosen to print the selected screen we remove the function selection menu strip right away so as to eliminate a host of MENUPIK messages which might be caused by the user banging on the mouse while waiting for the printout to finish.



Finally, if the user has selected the "screen-to-front" command we bring the selected screen to front. Then we wait for ANY message coming in over the IDCMP. If the message is an INACTIVEWINDOW the user elected to make the new front screen the active screen by clicking the left mouse button.

Any other message will cause scripper to bring its own screen (usually the workbench) back to the front thus implementing the quick screen toggling mechanism.

Notice that even at this point we check for the CLOSEWINDOW message and pass back a FALSE value if it should take place. This way, no matter what state scripper is in (besides printing) it will respond correctly to close gadget strikes.

Phewie!!! This has been a long adventure in C programming on the Amiga. In this three part tutorial we've covered the in's and out's of many advanced features in programming this exceptional machine. We covered C programming techniques and discussed why things might be done in one way over another.

And lastly, we've discussed some of the gestalt of programming. That is, you've been taken on a tour of algorithms, data structures as well as been given a glimpse of what goes on inside the head of a person whose been using C exclusively for seven years.

Maybe this code wasn't the best choice for a brain dump considering that when it was written (fully eight months ago) it was intended as a quickie exercise never to be seen by other humans. In my opinion, one of the most useful attributes of the scripper articles was precisely the fact that the code was not all that hot.

That is, for a change readers got to see a discussion of real world code rather than finely crafted mistake free code which does not lend itself to a discussion of alternative approaches.

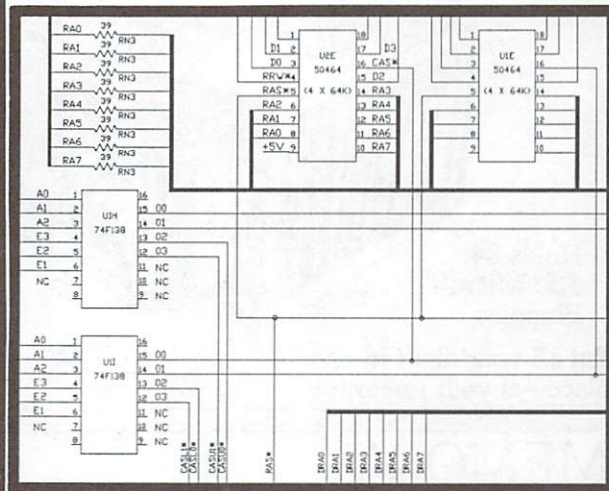
All in all, I hope that this has been a worthwhile contribution to your ramping up upon the Amiga learning curve.

In the future, I am considering changing the format of Miga-Mania to be the discussion of a different short and useful program every month rather than its current grab-bag style.

•AC•

## INSIDE YOUR GREY BOX

# AMIGA SCHEMATICS



## You can investigate:

- RAM Expansions
- Auto Boot Rom Mods
- Disk Drive Interfaces
- Additional Ports
- DMA Expansions
- Video Enhancements
- Etc. . .

# \$24.95

Orders Toll Free 10-6 MON-SAT

# 800 762-5645

Cardinal  
Software



14840 Build America Dr.  
Woodbridge, VA 22191  
Info: (703) 491-6494

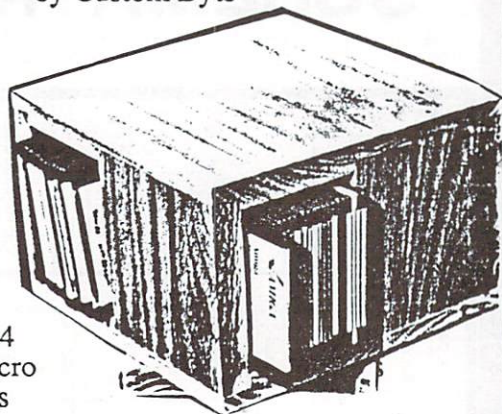


4130 N. 75th Ave. #105  
Phoenix, AZ 85033

DEALER INQUIRES WELCOME

NEW  
Amiga®  
Products!

## by Custom Byte



Holds 84  
3.5" Micro  
Floppies

Put all your disks in one place - at your fingertips

\$49.95

A BIG  
2 mb.

Features: Speed  
Auto configuring  
Mounts on expansion port  
Zero wait states

A must for business, graphics, database, and cad programs. **\$795.00**

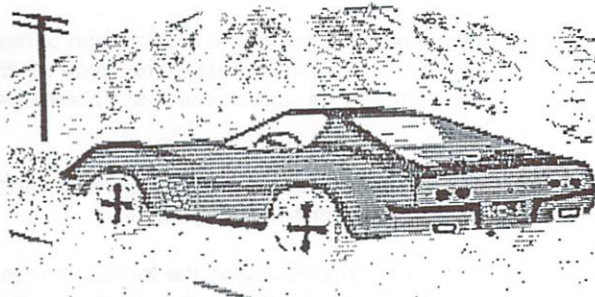
Gallery of Images For DELUXE PRINT®  
Chapter 1 



Featuring:  
States, Social Activities,  
Southwestern Art, Mix  
and Match

only **\$19.95**

Computer Art Gallery of Animations  
by Ken Costello



Corvette example; wheels spin, telephone poles  
flash by and road moves. 23 animations,  
Explore the animations of  
your computer. **\$19.95**

**\$19.95**

- |                                                                                                                                                                                                                                                                                                                                                                                                                               |       |                                          |          |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|------------------------------------------|----------|
| <input type="checkbox"/> Art Disk Deluxe Print                                                                                                                                                                                                                                                                                                                                                                                | 19.95 | <input type="checkbox"/> Memory Board    | \$795.00 |
| <input type="checkbox"/> Art Disk Deluxe Paint                                                                                                                                                                                                                                                                                                                                                                                | 19.95 | <input type="checkbox"/> Oak disk holder | \$69.95  |
| <input type="checkbox"/> check/money order <input type="checkbox"/> Credit Card No. _____<br><div style="display: flex; justify-content: space-between;"> <div>Exp. date _____</div> <div>Sign. _____</div> </div> <div style="display: flex; justify-content: space-between;"> <div><input type="checkbox"/> Visa   <input type="checkbox"/> M/C   <input type="checkbox"/> Amer. Exp.</div> <div>Computer West</div> </div> |       |                                          |          |

\$3.00 shipping and handling  
AZ residents add 6.5% sales tax  
(602) 849-4795

Mail to: 4130 N. 75th Ave. #105  
Phoenix, AZ 85033

Name \_\_\_\_\_  
Address \_\_\_\_\_  
City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_

Amiga is a registered trademark of Commodore Business Machines. Dextreme Print is a registered trademark of Electronic Arts.



# The Amazing C Tutorial

By **Stephen R. Pietrowicz**

CIS [73047,2313]

Usenet: ...lihn4!pur-ee!gould!houligan!srp

People Link: DR RITZ

In this lesson, we'll be exploring structures and unions, and how to use them in your own C programs.

Why use structures? Structures are used to consolidate information about a given "object". That "object" can be a page in an address book, an entry in an inventory list, a car in a parking lot... any object that has different pieces of information that you want to keep together in a single entity.

In Pascal, this is called a 'record'. In BASIC, you would use FIELD statements to define the layout of the data to be stored on disk.

Those pieces of information are kept in a form that is easy to access and manipulate. Structures are used by many Amiga system calls to handle information for the information the Amiga keeps internally. Information for creating windows, storing sprites and bobs, and creating gadgets are all stored in structures. You can create structures in your own programs to efficiently hold any type of information you choose.

## Declaring your own structures

Imagine that you have a mailing list. Each person on that mailing list has his own information: contact name, company name, address, city, state, zip code, age, and the number of mail items that have been sent to that person.

In a language such as Basic, each of these items would have to be kept in a separate array. Each element the array corresponds to a certain person's information. In C, this information can be kept together in a single structure:

```
struct Entry {  
    char *Contact;  
    char *Company;  
    char *Address;  
    char *City;  
    char *State;  
    char *Zip;  
    int Mailings;  
};
```

The tag of this structure declaration is Entry. For those of you who program in Pascal, this should look familiar. The Pascal equivalent to a structure is called a 'record'.

Declaring MailEntry structure variables is simple:

```
struct Entry Person1, Person2, DataBase[500];
```

'Person1' and 'Person2' each have enough room reserved in them to hold information about one entry in the mailing list.

'DataBase' is an array that can hold up to 500 different entries.

[Editor's note: Please note that the 'DataBase' array elements only hold pointers to character strings. The 'DataBase' array does not hold the string values themselves! Each array element is a structure of type 'Entry'. Each 'Entry' struct holds only pointers to character strings.

Look at the example below, at the line 'Person1. Contact = "Amiga"'. In this example, the pointer-to-character element 'Contact' is initialized to point at the string "Amiga".

In other words, the pointer is set to the address of the string "Amiga", which is a constant string compiled into the program. You can print this value, using the 'printf()' function. However, if you want to store this database to disk, saving only the structures in 'DataBase' itself would make little sense, since each array element only contains pointers to the data, not the data strings themselves. In this sense, this is not a practical example of declaring an array for a simple database program in C.

There are a lot of complications with using strings in C. It is not as easy as in BASIC or FORTRAN. As usual, C gives much more flexibility in dealing with the content of data. [The topic of C strings will be covered in a future tutorial.]

Assigning values to fields is also straight forward:

```
Person1.Contact = "Amiga";  
Person1.Company = "Commodore Bussiness Mach.";  
Person1.Address = "1200 Wilson Drive";  
Person1.City = "West Chester";  
Person1.State = "PA"  
Person1.Zip = "19380"  
Person1.Mailings = 6;
```

In a sense, we've just declared our own data type. It contains all the information we wanted to keep in a single definition. The open and close curly braces enclose the contents of the structure. Each of the structure 'elements' must be of any declared type: int, char, float.....even another structure type:

```
struct Account {  
    char *AccNumber;  
    float Balance;  
};
```





## CHECK US OUT!

MORE THAN A  
MAGAZINE,  
we're the largest  
Commodore Users  
Group in the world!

Take a look. Here's what you'll get...

- Access to library for C-64, Amiga, MS DOS, VIC 20, PET/CBM, Plus/4, C-16, SuperPET, B-128, C-128.
- 10 magazines featuring articles and advice from noted experts like Jim Butterfield, Steve Punter and Liz Deal.
- Access to the TPUG Bulletin Board System, a Punter BBS based in Toronto, Ontario.
- Free admission to TPUG chapter meetings, covering many special interest areas of Commodore computing.
- The right to attend TPUG's Annual Conference, with guest experts from around the world.

### TPUG yearly memberships:

Regular member (attends meetings) — \$35.00 Cdn.

Student members (full-time, attends meetings) — \$25.00 Cdn.

Associate (Canada) — \$25.00 Cdn.

Associate (U.S.A.) — \$25.00 U.S./\$30.00 Cdn.

Associate (Overseas — sea mail) — \$35.00 U.S.

Associate (Overseas — air mail) — \$45.00 U.S.

Associate Club (15 members) — less \$5.00 per person on appropriate membership.

FOR FURTHER INFORMATION: Send \$1.00 for an information catalogue or telephone: (416) 445-4524 (Please tell us which machine you use!)

TPUG INC. DEPT. 200

101 DUNCAN MILL RD., SUITE G7, DON MILLS, ONTARIO, CANADA M3B 1Z3

```
struct BankEntry {
    char *Name;
    struct Account Savings;
    struct Account Checking;
} Accounts[3000];
```

Notice that the Account structure doesn't have a variable declared; it is simply a declaration of a new data type containing one character pointer, and a floating point number.

It isn't necessary for the structures to appear in this order. If the BankEntry structure was declared before the Account structure, the compiler will assume that the Account was going to be defined later in the program. When you use a structure in a C declaration that hasn't been defined yet, but is defined later in the declarations, it is called a 'forward reference'.

Assigning values to fields of structures inside of structures is simple too:

```
Accounts[0].Savings.Balance = 1872.45;
Accounts[0].Checking.Balance = 340.23;
```

One of the advantages of using structures is that they can be acted upon as one unit. For example, instead of copying each individual field from one structure to another, you can copy all the fields simultaneously with one assignment:

```
Person2 = Person1;
```

### Automatic Initialization

C allows you to automatically initialize variables when you declare them. For instance, if you declared an integer and want to initialize it to 500, you declare it like this:

```
int Number = 500;
```

The same thing can be done with a structure. Suppose you have declared an Entry structure variable as described above, and you want to initialize each of the fields to a certain value. You can do that using a simple C assignment statement:

```
struct Entry Person1;
Person1.Contact = "Mary McGregor";
Person1.Company = "ABC Computers";
Person1.Address = "1234 Main Street";
Person1.City = "Hometown";
Person1.State = "IN";
Person1.Zip = "47906";
Person1.Mailings = 2;
```

However, using automatic initialization, the variable could have been declared like this and achieved the same result:

```
static struct Entry Person1 = {
    "Mary McGregor",
    "ABC Computers",
    "1234 Main Street",
    "Hometown",
    "IN",
    "47906",
    2};
```

Automatic initialization can also be done for arrays of structures:

```
static struct Entry DataBase[500] =
{ "Zero", "Zero", "Zero", "Zero", "Zero", "Zero", -1 },
{ "One", "One", "One", "One", "One", "One", -1 },
{ "Two", "Two", "Two", "Two", "Two", "Two", -1 };
```

This initializes the first three array elements of DataBase. Any array elements that are not assigned values are automatically assigned the value zero.

When using automatic initialization on structures that contain structures, the statement must contain the curly braces around each structure.

```
static struct BankEntry Person3 =
{ "Mr. Megabucks",
  { "12345", 245056.20 },
  { "67890", 16548.34 }, };
```

This statement is equivalent to:

```
struct BankEntry Person3;
Person3.Name = "Mr. Megabucks";
Person3.Savings.AccNumber = "12345";
Person3.Savings.Balance = 245056.20;
Person3.Checking.AccNumber = "67890";
Person3.Checking.Balance = 16548.34;
```



### Allocating Memory for Structures

Declaring pointers to structures is done in a similar way to declaring pointers to characters, or integers:

```
struct Entry *Ptr;
```

In order to assign values to individual fields when using structure pointers, the memory needed to store the structure must first be requested from the system. The system call to request memory from the system is called 'AllocMem()'. One way to request memory using this function is:

```
Ptr = (struct Entry *) AllocMem (sizeof(struct Entry),  
MEMF_PUBLIC);
```

Now, the first time I saw the equivalent UNIX system call, I was pretty confused. Does the function call have two parameter lists? No, the first part of the assignment, "(struct Entry \*)", casts the AllocMem function call into a pointer of type struct Entry. In other words, the return value from AllocMem is temporarily changed into an Entry pointer value so the compiler doesn't complain that you are trying to assign values to variables with conflicting types.

The 'sizeof' operator in the first argument to AllocMem calculates the size of the Entry structure in bytes. sizeof can be used with any declared type as it's argument, and is an easy way to calculate the size of those types. It's especially helpful when used with very large structures.

Size of is also helpful if you are writing portable C code. The size of data types may vary from machine to machine. By using the 'sizeof' operator, you can eliminate the machine dependency.

Of course, you can decide just to substitute that entire expression with 'sizeof's value, but you'll lose the portability. And if you decide to add another field to the structure you are using you will have to change that number in all of the AllocMem's you use in your program.

AllocMem's second argument is MEMF\_PUBLIC, and is one of four possible attributes that describe traits associated with allocated memory:

**MEMF\_PUBLIC:** Memory that must not be made non-addressable. Memory that is referenced by other tasks must public, this includes code and data.

**MEMF\_CHIP:** Memory that can be used by the DMA circuitry. (ie. Disk I/O, graphics operations using the Blitter) MEMF\_CHIP memory is taken from the first 512K of RAM.

**MEMF\_FAST:** Memory that is not in the first 512K of RAM. This type of memory should be used in non-DMA critical applications.

**MEMF\_CLEAR:** MEMF\_CLEAR should be OR-ed with one of the other three attributes. Memory returned will be initialized to zeros.

You can 'or' in the MEMF\_CLEAR value to tell 'AllocMem()' the newly allocated memory to zero.

## SHARPEN YOUR IMAGE With Digital Color Slides Posters

For Professional Presentations,  
Art Portfolio or even for Fun!

Let your Amiga images shine with the quality you deserve. Any image created from Deluxe Paint, Graphicraft or Propaint can be made into high quality Digital\* 35mm Slides or Studio Posters

No additional software or hardware needed. Just send us your files as they are stored on disk and in 2-3 days (plus delivery) you'll get back the proud results. Slides are \$13 each. Matte or gloss Studio Print Posters 11 x 17 are \$25.50 plus slide. Also available, Digital Color Separations (as seen in AmigaWorld Magazine) and 8 x 10 Color Studio Prints and Transparencies. Orders must be prepaid (with sales tax in California.) Send your disk and check to:



**ImageSet**  
corp.

555 19th St. 2nd Flr.  
San Fran., Ca. 94107  
415 626-8366

\*Images are not captured by photographic methods.

## Rescue Your Recipes!



## Compu Cuisine

Over 200 Recipes  
in 8 categories.

Edit and add own recipes.

Organize and search files  
by category or ingredient.

Send check or money order for

\$29.95 to:

*for the Amiga!*

Adept Software

P.O. Box 700702

San Jose, Ca. 95170





# COMPUTERS

WE'VE  
**SLASHED  
PRICES**  
ON THE **AMIGA!**



**The West Coast's Largest Inventory of Commodore And Amiga Software And Hardware Products**

**CALL KJ (818) 366-5305 • 366-9120**

10815 Zelzah Avenue Granada Hills, CA 91344

'Ptr' is assigned the return value from the call to 'AllocMem()' which is the address of the first word in the memory block requested.

Structure assignments using pointers are similar to regular structure assignments:

```
Ptr->Contact = "Amiga";
```

Note the use of ">" instead of "." as in the previous example.

### How structures are stored

When structures are stored in memory, their fields are stored in consecutive memory locations. When the memory is allocated for a structure as listed above, it is returned in a single memory block. When you reference a field in that structure, C automatically indexes into that memory block to retrieve the location that you wish to use. It works similarly to indexing in an array. Ptr is the address of the beginning of the block, and by adding the correct offset, you can index to the correct variable in the structure.

For example, when the expression "Ptr->Contact" is encountered, C converts this to "Ptr+0", the address of Ptr plus an offset of zero. In the same way, "Ptr->City" is converted to "Ptr+12", or the address of Ptr plus an offset of 12.

Fields of structures are 'word aligned'. Instead of packing fields in structures to reduce the size if possible, C leaves left over space in some fields. If the size of a variable is smaller than a word, the rest of the word is wasted space.

For example:

The structure:

```
struct a {
    char a, b;
    int c;
    char d;
} sample;
```

Is stored like this:

| Offset: | Field:            |
|---------|-------------------|
| 0       | used   wasted   a |
| 2       | used   wasted   b |
| 4       | fully used   c    |
| 6       | used   wasted   d |

### Unions

Unions are similar to structures, but only one field can be present at any given time. In other words, the union:

```
union {
    int A;
    char B;
} temp;
```

can only hold one variable at a time, either the integer A, or the character B. (Accessing individual fields within the unions is the same as if you were using structures).

This might not sound particularly useful. Why not just declare a structure and have one field for A and one field for B?

When C encounters a union construct, it scans the fields within the union for the largest field in that union, and reserves that amount of space. Normally unions are used when you know that you're only going to need to store one type of variable in your data structure at a time, and you need (or want) to save space. In Pascal, the equivalent of unions are 'variant records'.

For example, in an inventory for a store some items are measured in pounds, (flour, sugar, etc), others are measured by the quantity in the store, (fruit, vegetables, etc). A structure can be set up to describe this:

```
struct Item {
    char *Name;
    float pounds;
    int quantity;
};
```

and depending on what the value of the Name field is, the program can pick either pounds or quantity. Alternatively, the structure could have been set up this way:



```

struct Item {
    char *Name;
    union {
        float pounds;
        int quantity;
    } num;
};

```

Note that you are still able to determine which field in the union to pick by scanning the Name field, and you save space in the structure. In a large scale application, intelligent use of unions can save quite a bit of memory.

Because of the way unions are stored, you can perform operations in C that you normally can't do. For example, you have to write a function to convert a floating point number from the way the Amiga stores it internally, to the internal representation of another type of computer.

In order to do that, you would have to perform bit manipulation on the floating point number. However, it is illegal to perform bit operations on floating point numbers in C. The compiler simply will not let you do it. You can however trick C into allowing you to do it by using a union:

```

union {
    float a;
    int b[2];
    char c[4];
} d;

```

By assigning the number to split up to D.a, we can then look at D.b[0] and D.b[1] to see the high and low order words in the number, and D.c[0], D.c[1], D.c[2] and D.c[3] to see each byte of that number.

With what you've learned so far in the Amazing C tutorials, you should be able to begin experimenting with C to write your own Amiga programs. In future C tutorials, we'll explore the Amiga system calls and how to use what you've learned so far to program your Amiga.

•AC•

## 256 K ADD-IN MEMORY

► 95\$ ◀

If you bought your Amiga without the 256K additional memory and have \$95, this is the deal you have been waiting for! Send a check (certified for faster processing, at this price the cash flow situation is very tight) to the address below. If you already are running 512K, or are not sure what to do, send \$1 for a listing of currently available products for your Amiga.

(Prices include shipping)



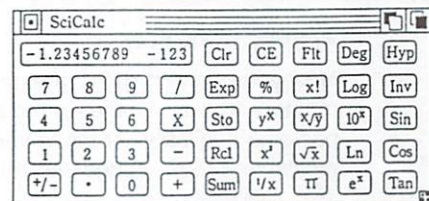
**NIKOL & COMPANY**

*Computers*

2545 Bainbridge Blvd., Chesapeake, Virginia 23324

## SciCalc™

### Scientific Calculator For The Amiga™



**\$14.95**

Don't let the price fool you! SciCalc has full algebraic hierarchy and features an automatic constant that is a delight to use. Choose from 3 display modes: Floating Point, Scientific, or Fixed Point..

Press the Hyp(erbolic) key twice and a whole new page of functions is at your fingertips. No long waits - SciCalc has been available since March. Your order with manual will be sent by First Class mail.

#### Features

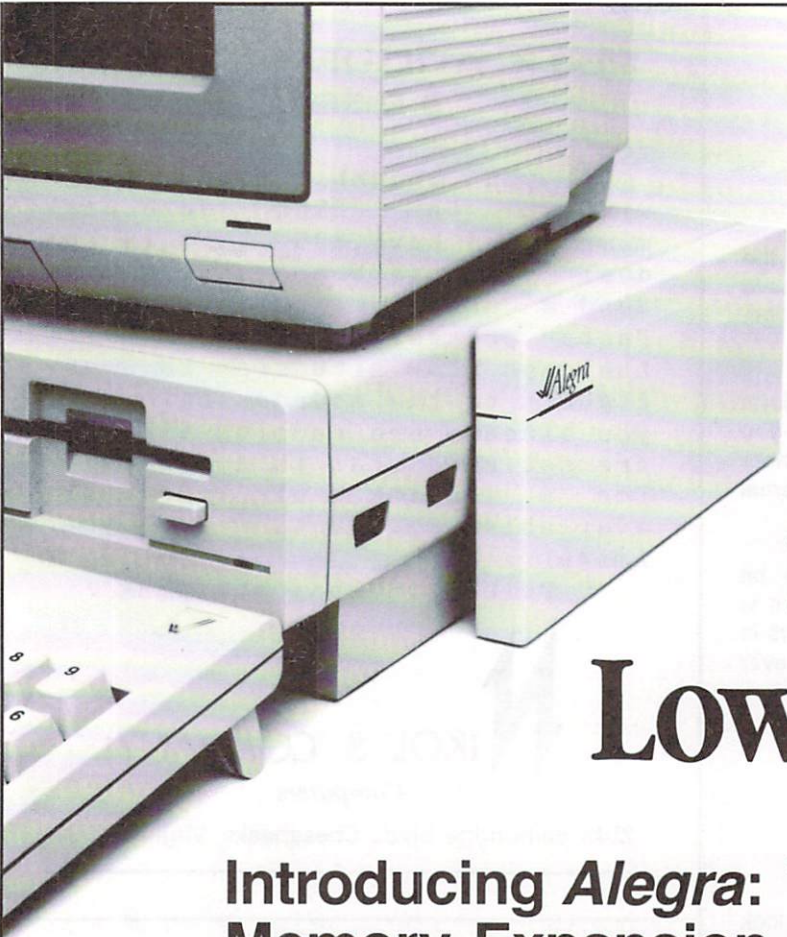
- Large Equals Key (Display)
- Adjustable Size
- 10 Memories
- Powers
- Logarithms
- Trigonometry (D/R/G)
- Hyperbolics
- Color Highlighting
- Full Error Trapping
- 2 Dimensional Statistics
- Linear Regression
- Linear Estimation
- Correlation Coefficient
- Factorials to 170
- Polar/Rectangular Conversions, and more.

#### Dealers Inquiries Welcome

Send Check for \$14.95 to:

**DESKWARE**  
P.O. Box 47577  
St. Petersburg, FL. 33743





# Smallest footprint. Lowest price.

## Introducing *Alegra*: The Amiga™ Memory Expansion Unit from Access Associates.

### ***Alegra*: 512K now.**

Now you can add 512K of external memory to your Amiga. In the smallest package available. At the lowest price available: \$339<sup>00</sup>, suggested retail.

### ***Alegra E*: 512K now. 2 Mb later.**

If you'll need 2 Mb of memory in the future, the upgradeable *Alegra E* is the right choice now. *Alegra E* gives you 512K bytes of added memory with the option of 2 Mb later.\* Because the upgrade requires only internal component changes, *Alegra* maintains its slim 3/4"-wide footprint. The price: \$379<sup>00</sup>, suggested retail.

*Alegra* features a 90 day parts and labor warranty against manufacturing defects.

See *Alegra* at your quality Amiga dealer.

Dealer inquiries invited.

### **ACCESS ASSOCIATES**

491 Aldo Avenue  
Santa Clara, CA. 95054-2303  
408-727-8520

\*As 1 Mbit DRAMs become widely available.

\*\*Amiga is a trademark of Commodore Amiga, Inc.



# Optimize Your AmigaBasic Programs For Speed

By **Stephen R. Pietrowicz**

People Link: Dr: Ritz

CIS [73047,2313]

Usenet: ...lihn4!pur-ee!gould!houligan!srp

You've just written an AmigaBasic program, and you're not satisfied with it's speed. What is the best way to go about making the program run more quickly?

By following some techniques used by compilers to optimize programs for speed, and by using a little bit of common sense, you'll be able to squeeze a few extra seconds out of that program!

First we'll cover a few different techniques, and then apply these optimizations to some examples at the end of this article.

Compilers for C, Fortran, Pascal, and other languages use a variety of techniques to optimize code for speed. After generating machine language code, some compilers scan the object code and decide where to optimize.

Some compilers do a flow analysis of where the program spends most of it's time, and how variables are updated within loops in the program. The compiler can then make "intelligent" decisions about what to change in your program to optimize it.

The optimization techniques discussed below can be applied to almost any language at the source level. Most compilers perform optimizations after creating the machine language equivalent of a program.

Probably the simplest optimization method is known as "constant folding". The compiler scans the code for constants that it can combine during compilation, and thus eliminates the calculations during execution. Consider the following statement:

```
x = 11 + 2 * (x * 9 / 6 + y) * 8.6 + 4 * 5
```

A compiler performing constant folding would change the statement to this equivalent statement:

```
x = 17.2 * (x * 1.5 + y) + 31
```

It eliminated two multiplication, one division, and one addition operation. While most programmers would not leave constants like those above uncombined, we will see in the examples later that we can do the above to reduce the number of mathematical operations in some time critical loops.

Another common optimizing technique is "code motion". The compiler scans for expressions that never vary within loops, and moves the code outside of the loop. For example:

```
FOR I = 1 TO 360 STEP 10
  FOR J = 0 TO 360 STEP
    R = 20 + W
    K = SIN(I) * SIN(J) + COS(I) * COS(J)
    Z = Z + J + R
  NEXT J
NEXT I
```

The obvious statement to move is "R = 20 + W" in the inner most loop, because nothing in either of the loops changes the value of R or W. The compiler would then move the statement completely outside both loops.

A less obvious "code motion" optimization can also be done, and will eliminate many more calculations. Since the variable "I" remains constant within the inner loop, so do the values of SIN(I) and COS(I).

It is therefore possible to move those two calculations to the body of the outside loop. The result of the code motion optimizations is below:

```
R = 20 + W
FOR I = 1 TO 360 STEP 10
  X1 = SIN(I)
  X2 = COS(I)
  FOR J = 1 TO 360
    K = X1 * SIN(J) + X2 * COS(J)
    Z = Z + J + R
  NEXT J
NEXT I
```

Moving two SIN and COS function calculations out of the inner loop will significantly increase the speed of the inner loop's execution by reducing number of calculations made.

When the compiler changes the first assignment in the inner loop, it generates it's own internal variables, X1 and X2, and replaces the SIN(I) and COS(I) those variables. You should note the trade-off involved - this optimization uses more memory.

You can use these methods plus others to optimize your own programs. Since AmigaBasic is an interpreted language, and has no built in optimizer you'll have to decide where to change your code. Here are a few more tips to keep in mind when you're trying to optimize your programs:



# Amiga Project

## Programming Journal for the Amiga

A no-nonsense journal dedicated to programming for the Amiga, on the Amiga. Monthly columns written by experts covering Forth, C, assembly, Modula-2, Pascal, and more.

Help and advice for those of you who want to REALLY program your Amiga.

*No HYPE, No Gee-Wiz!!!!*

\$24.00 for 12 informative issues....

Send your check or money order to

**Amiga Project**

P.O.Box 285

Kent, OH. 44240

216-673-0185

Dealers, Advertisers, Call for current pricing info...

**1) Avoid using variables when you can use constants.**

Whenever AmigaBasic scans a variable in a program, it has to look it up in its table of variables for its value. Look ups don't take very much time, but if your program does many calculations, the time adds up quickly.

**2) Do not put comments in the main loops of programs.**

Scanning past comments doesn't take AmigaBasic very long, but putting them in loops adds to the time required to execute the loop. This doesn't mean never comment your programs! You should ALWAYS comment your programs! Commenting your code will make it easier for you to go back to later, especially if you've done something particularly tricky. It also makes it easier for others that will see your program and will try to understand it.

**3) Know all the functions in AmigaBasic. This may sound simple, but it is very easy to over look some of the functions that AmigaBasic has to offer. Why reinvent the wheel?**

**4) Avoid calling subroutines too often.** If you have a program that constantly loops and calls the same subroutines repeatedly, move the code for the subroutines inside the loop. If the subroutines are large, consider combining all the subroutines into one. Each time you call a subroutine, AmigaBasic keeps track of where to return to when it is finished executing the subroutine. Doing this over and over slows down your program.

Let's take a look at an AmigaBasic program to see how to go about doing an optimization.

Listing 1 contains three different subroutines to plot a simple mathematical function 5 times using different values for each of the plots. Our goal is to get the plots to the screen as quickly as possible.

The original subroutine is called SlowPlot, and takes a little over twominutes to run. Applying the guidelines listed above, we see we can immediately eliminate three "slow spots" in the subroutine:

1) Take out the comments, or move them out of the inner loops.

2) Take out the FN a functions, and combine the constants.

3) Merge the duplicate calculations into one calculation and use that.

By watching the plot, we see that the left side of the plot is the same as the right side of the plot. We can take advantage of this symmetry, combine the inner loops, and plot both sides of the function at the same time. By combining the two inner loops, we've just eliminated half of the calculations needed to draw the plots.

The changes made to SlowPlot result in the second function, QuickPlot. QuickPlot takes about 45 seconds to run. Quite an improvement! Let's take a look at what we can do to QuickPlot to improve its speed.

Notice that the only variable that changes in the inner loop is the variable "j". The calculations done with "x" repeat each time. By exchanging the inner and outer loops calculations done with "x" are done only once for each point, instead of five times for each point.

The last subroutine, QuickerPlot, makes this improvement, and takes about 35 seconds to run. The subroutines are arranged in listing one so you can run them and observe the execution times for yourself.

The last optimization technique I'll explain is absolutely the best, but it is also the most difficult to do. Once you've done absolutely everything to improve the algorithm that you've been working on, "step back" mentally, and look at your program with a very critical eye. Is the routine you've written REALLY the best one that you can come up with? This is extremely difficult to decide, especially for beginning programmers.

Here's a little test for yourself: right now, think how of you would design a routine that shuffles a deck of cards. Assume that an array is already initialized with the names of the cards, and that you can use any additional variables that you want to use. Don't take too long, just use the first idea that comes to you. When you're done, continue reading.

Most programmers would come up with something similar to the routine ShuffleA shown in listing 2. The idea is this: Create two more arrays, one to put the cards into after you've picked them, and another to mark each card after you've used it. (Remember, you don't want to have duplicates!)



First clear the "marked" array. Set up a loop to fill each space in the "shuffled" array. Now, use the random number generator to pick a random card, and check the "marked" array to see if that card has already been chosen. If it hasn't, put the card in the next place in the array. If it has, pick another random number, and repeat until you're filled the entire array.

While this isn't really a bad algorithm, it does have the potential for taking a long time to execute. This is what could happen in the worst case: Randomly place the first card into the "shuffled" array. The next card, and each card thereafter, could be a repeat of one of the previously picked cards.

You might have to continue picking random cards for quite a while to come up with one you haven't used. What should you do? It's possible to build in some kind of "artificial intelligence" into the algorithm, but this is really more trouble than it is worth.

ShuffleB, also listed in listing 2, is a better solution. It doesn't use any additional memory beyond the array that has already been initialized, and it always runs at the same speed.

The idea is actually quite simple. For each of the cards in the array, randomize a number between 1 and 52, and swap the current card with the card you've just randomly selected. The routine only has to pick 52 random numbers.

To show the difference in speeds, I set up both routines to shuffle an initialized "deck" 10 times each. ShuffleA takes between 15 and 18 seconds to run. ShuffleB consistently takes 6 seconds to run.

I told you to use the first algorithm that came to mind for good reason. Too many programmers use the first idea that comes to mind, and don't even try and think of another way to solve a problem. I can't over-emphasize the importance of getting a good algorithm. Always try to think of different ways to program a problem; you might just stumble onto a better, and quicker, solution!

#### Listing 1

```
' Optimize Your AmigaBasic Programs For
Speed
' By Stephen R. Pietrowicz
' Listing one - Plotting example
```

```
DEF FNa(t) = (100/360*t)/3

' Draw each of the plots one at a time
' and report all times when done

CLS
GOSUB SlowPlot
FOR i = 1 TO 10000:NEXT i

CLS
GOSUB QuickPlot
FOR i = 1 TO 10000:NEXT i

CLS
GOSUB QuickerPlot

LOCATE 1,1
```

## AMIGA CUSTOM PRINTER DRIVER:\$35+S/H

Create your own printer driver for  
virtually any printer.

• MENU DRIVEN • WORKS THRU PREFERENCES

We are world famous for our selection  
of Amiga software!

Call our Amiga BBS at night or call  
during store hours to order!

We specialize in  
COMMODORE AND AMIGA COMPUTERS  
SOFTWARE SUPERMARKET  
31621/2 Delaware Ave.  
Kenmore, N.Y. 14217

For Dealers only:  
Please call for pricing  
on our printer driver.



(716)873-5321  
& THE PRINTER STOREhouse

```
PRINT "SlowPlot:      Begin -- ";a1$;" End -
- ";a2$
PRINT "QuickPlot:    Begin -- ";b1$;" End -
- ";b2$
PRINT "QuickerPlot:  Begin -- ";c1$;" End -
- ";c2$
END

SlowPlot:
PRINT "SlowPlot"
a1$ = TIME$
FOR j = 20 TO 100 STEP 20
  FOR x = -150 TO -1 STEP .2
    ' First draw right side
    y = SIN(FNa(x))/(FNa(x))*j+50
    PSET (x+150,y),1
  NEXT x
  FOR x = 1 TO 150 STEP .2
    ' Now draw left side
    y = SIN(FNa(x))/(FNa(x))*j+50
    PSET (x+150,y),1
  NEXT x
NEXT j
a2$ = TIME$
RETURN

QuickPlot:
PRINT "QuickPlot"
b1$ = TIME$
FOR j = 20 TO 100 STEP 20
  FOR x = 1 TO 150 STEP .2
    sy = x*.0925926
    y1 = SIN(sy)/sy*j+50
    PSET (150-x,y1),1
    PSET (x+150,y1),1
  NEXT x
NEXT j
b2$ = TIME$
RETURN

QuickerPlot:
PRINT "QuickerPlot"
c1$ = TIME$
FOR x = 1 TO 150 STEP .2
```



```

sy = x*.0925926
SinTemp = SIN(sy)/sy
FOR j = 20 TO 100 STEP 20
  y1 = SinTemp*j+50
  PSET (150-x,y1),1
  PSET (x+150,y1),1
NEXT j
NEXT x
c2$ = TIME$
RETURN

```

#### Listing 2:

' Optimizing Your AmigaBasic Programs for Speed

' Stephen R. Pietrowicz

' Listing 2 - Card Shuffling example

```

DIM card$(52), Shuffled$(52), Check(52)
DIM Spot$(4)

```

```

Spot$(0) = " of clubs"
Spot$(1) = " of spades"
Spot$(2) = " of diamonds"
Spot$(3) = " of hearts"
DATA
"ace", "two", "three", "four", "five", "six", "seven"
DATA
"eight", "nine", "ten", "jack", "queen", "king"

```

```

Main:
RANDOMIZE TIMER
' ShuffleA
PRINT "Shuffling...."
a1$ = TIME$
FOR S = 1 TO 10

```

```

RESTORE
GOSUB Init

```

```

FOR I = 1 TO 52
  Check(I) = 0
NEXT I

Again:
FOR I = 1 TO 52
  X = INT(RND*52+1)
  IF (Check(X) = 1) GOTO Again
  Shuffled$(I) = card$(X)
  Check(X) = 1
NEXT I

NEXT S
a2$ = TIME$
PRINT "ShuffleA -- Start: ";a1$;" End ";a2$
' Shuffle B
PRINT "Shuffling...."
b1$ = TIME$

FOR S = 1 TO 10

  RESTORE
  GOSUB Init

  FOR I = 1 TO 52
    X = INT(RND*52+1)
    SWAP card$(I), card$(X)
  NEXT I

  NEXT S
b2$ = TIME$
PRINT "ShuffleB -- Start: ";b1$;" End ";b2$
END

Init:
X = 1
FOR J = 1 TO 13
  READ A$
  FOR I = 0 TO 3
    card$(X) = A$+Spot$(I)
    X = X + 1
  NEXT I
NEXT J
RETURN

```

•AC•

## Amazing Writers!!!

Yes, we mean you! If you enjoy Amazing Computing and you are using your Amiga, you have completed one half of the qualifications of an Amazing Writer for Amazing Computing™.

We are interested in the tasks and joys you have experienced on the Amiga. We want to read the secrets you have unlocked. We want to experience your excitement and enthusiasm. If you own an Amiga, you have already qualified as an independent thinker, now use that ability to communicate your individual story or idea.

Amazing Computing™ pages are filled with people who want to reach you with their thoughts. They explain a portion of the computer you both use and abuse, because they found it interesting.

If there is something in the Amiga family that interests you, chances are there are people who would enjoy hearing what you have to say. So don't sit around waiting for others to teach you what you have already learned by hours of trial and error, get

excited and teach the rest of us.

If your idea or explanation is of interest to developers and hard core hackers, please send your thoughts and a request for writer's guide lines to: AMICUS Network Editor.

If you are more interested in general use of the Amiga and its products, please send your suggestions and ideas to: Editor, Amazing Computing™

But, either way post them to: PiM Publications Inc.  
P.O. Box 869  
Fall River, MA 02722

Please include a hard copy and an electronic copy of your article for review. In both instances, please include your name, address and phone number. We will return an answer as soon as our editors stop shouting about how great your idea is, and types a response.

**Amazing Computing™: your resource to the Commodore Amiga**



# The Amicus Network

## Telecommunications Tips

By John Foust

Compuserve [72237,135]  
People Link AMICUS  
Delphi JOHNFOUSTuucp  
through the Well "jfoust"

The fastest way to get the latest public domain software is on a computer network or local bulletin board. In a sense, the national networks are a meeting place for national user Amiga groups. They aren't as formal as a local user group, and members come and go at will. However, you can meet with someone from across the country more often than you attend your local user group meeting! By honing your telecommunication skills, you can get the latest programs for the Amiga with little trouble.

There are a few skills you need to learn before grabbing your fill of free or nearly-free software. This column will discuss a few of the problems of sending files around the country, and across town, and explain the reason why 'arc' and 'chop' have become popular Amiga programs.

As with many things, there is an easy way to do things, and a hard way. Unfortunately, the hard way to transmit files is Xmodem, and it is the defacto standard. It works well with IBM PC computers, so few people are willing to change it. It has several limitations, and one of these collides head-on with the Amiga.

If you visit a local bulletin board, or subscribe to a national network, visit the public domain file area. If you don't have telecommunications experience, you might have trouble getting a copy of the files you see there.

There are several pitfalls in transferring files with your modem. The first is learning the proper commands to access the public domain files on the computer you visit. It may be best to leave a message there, if you have any questions, to get proper assistance for that system. Most systems have menus for accessing different parts of the computer. After you are in the Amiga area, make menu choices that have the word "File" or "Library" in them, and you get to the file library soon enough. With a little background, this process of getting public domain files is made easy.

There are problems inherent to the medium. It is difficult to transfer data reliably across phone lines. In a voice phone call, you are not hampered by occasional static on the line. Static means lost bytes. For a computer, a little static means some transmitted data will be garbled. If you are reading text messages from other visitors to the network, this won't make a big difference. If one or two characters are garbled, you can still read the message. If you are transferring a program file, a scrambled bit or two means the program will not work when it gets to your computer.

File transfer protocols exist to allow error-free file transfers across phone lines. Since people often want to send binary and text files without errors, file transfer protocols are used for all file transfers, whether across the room, or across the country. Perhaps you've heard the name of one of these protocols.

The most common, by far, is 'Xmodem', developed back in the days when CP/M was the predominant operating system. Another is 'Kermit', more popular in the educational computer world. There are many more. The Compuserve network has its own file transfer protocol, 'CIS-B'. Each has its own advantages and shortcomings. In essence, each protocol breaks a file into pieces, and transmits each piece separately.

The protocol defines extra information that tags along with each packet. For example, it might include the packet number, so the receiving computer can put the file back together in the correct order. It also carries information to detect if the packet was garbled by noise on the phone line.

When the receiving computer gets a new packet, it checks the extra information. If it was expecting packet number 15, and it got packet number 16, it tells the sending computer that it missed a packet. If it detects a packet was garbled, it asks the sending computer to re-send the packet. When the correct-numbered packet arrives intact, the receiving computer tells the sending computer, and the sending computer sends the next packet.

### Handshaking across America

This system of handshaking is common to all protocols, in some form. It is similar to the process two people use when giving an address over the phone. As each part of the address is given, the other person acknowledges it was received correctly, or asks the person to say it again. In order to properly synchronize this transfer process, both sides must start the transfer at the same time. One side will wait for a 'go' signal, and then it will send the first packet.

Again, check with the computer's authorities to get the detail of this process. As with all tasks performed by computer, this process of checking each packet takes time, and this affects the time it takes to send a file without errors. Even though your 1200 baud modem can send 120 characters a second, or about 7 K a minute, a file-transmission protocol will reduce your effective transmission rate by about two-thirds, at best, or about 70 to 90 characters a second.



# INTERACTIVE ANALYTIC NODE

## THE EXPLORER

### A tool to match your curiosity!

**Would you like...** to scout the inner workings of your Amiga? ...a live window onto memory to watch what other tasks are doing? ...an on-line memory map to tell you where you are? ...to actually see the assembly language code, in human readable form, that exists inside your Amiga? ...to step through a piece of code to see what it does? ...to capture your own source code and customize it?

The EXPLORER has some powerful features that make it a superb extension of your curiosity. **Features:** display memory and files in Hex and ASCII, memory modify, search, move, fill, display and change registers, disassembly trace, load programs, disassemble to disk. Output to printer or disk file. Powerful commands: loops, text display, real-time RAM view, & more! The EXPLORER puts your sense of wonder in charge!

The EXPLORER can be used for serious program development too! As a debug tool the EXPLORER's command set is compact and efficient. You can execute your commands within loops, creating live displays of RAM or registers while you test your program. You control the display format too, and even display informative messages. No need to waste valuable time typing that patch into memory every time you debug. Simply write a new command to do it for you. After all, what are computers for? When you want to save the contents of RAM or a series of trace steps for future examination, just send them to the printer, or better yet, send them to a disk file!

**PRICE: \$49.95** plus \$3 shipping and handling.

COD add \$4. Visa/MC orders call (612) 871-6283. Money orders or checks to:

Interactive Analytic Node  
2345 West Medicine Lake Drive  
Minneapolis, Minnesota 55441

If the file is coming from a large mainframe computer across the country, it takes additional time for the information to travel through the phone lines. Each confirmation of error or receipt will take a few extra fractions of a second to travel back and forth. If you have lights on your modem to indicate data transmitted and data received, you can watch this process. Usually, these lights will be marked 'RD' and 'TD'.

All the large national network computers are multitasking, just like the Amiga, and they may be bogged down with other users. This severely affects the effective transmission rate. If the lights on your modem are staying off for as long as they are staying on, you know that these delays are slowing your transmission. When you are paying between \$4 and \$24 an hour to visit a national computer network, this sort of delay is unacceptable. If the transfer is taking too long, you should abort the transfer and try again later at night, when there is less of a load on the computer. For a local bulletin board, you shouldn't have this sort of problem. There will be little or no delay between acknowledgements or packet transmissions, so either the RD or TD light will stay on almost continuously.

### How long is too long?

How long should a file take to transmit? With the Xmodem protocol found in almost every commercial and public domain terminal program, the file is broken into packets of 128 bytes each, or eight packets to a kilobyte. With the Compuserve B protocol, each packet is 512 bytes, most often. It's easy to calculate the number of packets it will take to transmit the file. If the host computer says the file is 11834 bytes long, and you are using Xmodem, count eight blocks for each thousand bytes, since a kilobyte is roughly 1000 bytes. For every four K in the file, add another packet. So this file will take about  $88 + 3$  packets, or 91 packets. Ideally, at 1200 baud, each packet will take a little more than a second to transmit, since each Xmodem packet contains about 130 characters, counting the extra information with the 128 bytes of data from the file. Remember that a transfer on a network will take slightly longer than this.

An easy way to speed file transfers is to use the RAM: disk. The floppy drives will slow the reception or transmission of data. If you are sending a file, first copy the file to the RAM:disk. If you are receiving a file, enter the filename prefaced with 'RAM:', to send it to the RAM: disk instead of the floppy. With the Online! terminal program, this can save a lot of time, since it doesn't buffer the packets in memory. Some public domain terminal programs are smarter about this, and store the packets in RAM before sending them out the modem, or copying them to disk.

### 'chop' and 'arc'

Xmodem transmits files in 128 byte pieces. Unfortunately, Xmodem has no method of transmitting the true length of the file, it always rounds up the file size to 128 byte boundaries. If the file is 90 bytes long, then Xmodem adds padding characters to fill out the file to 128 bytes. Since Xmodem is such an old protocol, it has become very popular.

On a computer that runs the CP/M operating system, file size doesn't really matter for executable programs, since the



whole program is loaded into memory, and execution begins at a fixed spot in the file. Since IBM PC-DOS is just warmed-over CP/M, it doesn't care about rounding file sizes to 128 increments. So, for almost all applications on an IBM-PC, Xmodem is a fine file transfer protocol. The Amiga is more finicky than the PC.

#### Not an object module?

On the Amiga, file size does matter. An executable program can be loaded into any section of memory. An Amiga program file contains extra information about the memory it needs. When Xmodem adds the extra padding characters to the file, it confuses the operating system, and it can't load this file as a program. In the CLI, this gives the well-known "Unable to load XXXXX: file is not an object module" error message. Here's where the 'chop' program steps in.

When you enter the file library area of the computer, you can ask for a description of the files in the database. With each file, you should find a description of the file. Part of the description of the file should state the proper file size. When you download the file, record this number. Instead of a file size, the description could say something about 'arc', such as 'This is an ARC file.' The 'arc' program will be discussed later. If the program has been 'arc'hived, you won't need the 'chop' program, but you shouldn't skip this discussion. 'arc' files do not need to be chopped. 'chop' will remove the extra padding characters from a program file, but you need to give it the proper file size. That is why you should record the number from the description. The 'chop' program will make a new copy of the program, minus the padding characters. If you have downloaded an IFF picture file, there is no need to 'chop' the file. IFF files have the true file size stored within them, so the program you use to view or edit the picture will be able to load it correctly, regardless of the extra padding characters.

There could be a question of bootstrapping here, too. For example, what do you do if you want to download the 'chop' or 'arc' program - you have no way to chop that file, so this has become a chicken-and-egg situation. Fortunately, there is a program called 'hunkpad', which can treat a program file so that it will always load and run, regardless of file size. It adds the right kind of padding characters, instead of the garbage padding characters Xmodem uses. Smart networks and bulletin boards have 'hunkpad'ded the 'chop' and 'arc' programs, so there is no need to 'chop' them before using them. If your BBS hasn't done this, tell them about it.

As an alternative, there are 'chop' programs written in BASIC. BASIC files don't care about the Xmodem padding, either, so you can always get this file, and use it to 'chop' the 'chop' program. There is also an automatic 'chop'-type program, called 'fixobj'. This program doesn't need to know the proper file size, it is smart enough to look at the end of the file, and remove the garbage characters. Please note that the 'fixobj' program only chops program files. It won't work on binary data files, such as spreadsheet templates.

#### File compression

It is hard to imagine, but there is a lot of 'air' in a file. In a textfile, at least one or two bits is wasted for every byte in the file. There are about 95 printable ASCII characters all

# INTERACTIVE ANALYTIC NODE

## NEW! EXPERT SYSTEM KIT

Those of us who get excited about computers have been looking for the application that takes us into the twenty-first century. This is it! Now you can create an expert system that will grow with your Amiga. Would you like a computer system straight out of science fiction sitting in your own home or office? How powerful can it be? As big as your imagination, because you build it the way you want it!

We supply the EXPERT SYSTEM driver along with a sample knowledge base. Complete instructions guide you to creating your own application. Experiment with artificial intelligence! The software driver analyzes your data and learns to draw the correct conclusions. Think of the applications! Diagnose circuits, plant and animal diseases. Predict events based on past performance—weather, stock market, sports. Build the ultimate science project or develop a commercial application! We will be supporting this kit with a newsletter so you can share knowledge bases, techniques and ideas.

**PRICE: \$69.95** plus \$3 shipping and handling.

COD add \$4. Visa/MC orders **call (612) 871-6283**. Money orders or checks to:

Interactive Analytic Node  
2345 West Medicine Lake Drive  
Minneapolis, Minnesota 55441



# ProForma™--An Enhanced Text Processor

ProForma™ is a powerful text processor and formatter with roots in NROff, a Unix text processor. When used with an Amiga editor, ProForma™ becomes a professional document production tool. ProForma™ supports most popular letter quality and dot-matrix printers.

Vertical format control - page length, header format, window line, variable line spacing. Horizontal format control - margins and indent. Multicolumn print - up to 4 text columns per page. Fill and justification Tabular Data format Headers and footers - with page numbering. Character formatting - boldface, underline, italics, superscript, subscript, strikeout, double width, alternate fonts (up to 3).# Macro definition - create new ProForma™ commands. Change bar generation - for revision control. Table of Contents Generation Index Generation Document preview - formatted documents on the screen.

**Suggested Retail Price - \$75.00**

ProForma™ developed by Future Concepts, Inc., exclusively distributed by:

Professional Network Services Corporation  
315 Chestnut Street  
Needham, MA 02192  
(617) 449-6460  
Dealer Inquiries Invited

ProForma™ - a trademark of Professional Network Services Corporation \* - character formats and fonts vary with printers supported.

told, but a byte can represent 256 values. Also, there are repeated sequences of characters in a text file, words that are repeated, and the pair of line feed characters at the end of every line in a double-spaced document. Looking at it another way, there are repeated streams of bits in a file. It is possible to reduce the storage space used by a file by compressing these sequences of bits and bytes. The file can be squeezed of its 'air', and restored to its full size when it is needed. The best program for file compression is 'arc'.

On text files, 'arc' can reduce the size of the file by forty to fifty percent. Program files are less regular and more dense, so they can be compressed only ten to twenty percent. The advantages of file compression are obvious. If you are anxious to minimize the time you are spending on an expensive computer network or long-distance phone call, a file compression program can reduce your online transmission time by as much as half. You can store more files on a disk. This makes the most sense for archive disks. If you keep a transcript of your telecommunications sessions, you can squeeze them, and unsqueeze them when you are looking for a particular message. (In case you are wondering, this compression can't go on forever. Re-'arc'ing an 'arc' file doesn't make it smaller, since 'arc' already did the best job it could on that data.

Imagine any file as a very large binary number. Under a given translation scheme, it is possible that there exists a smaller binary number with the same information content, and this is the way 'arc' works.) Perhaps you've heard of the 'sq' and 'usq' programs, pronounced 'squeeze' and 'unsqueeze'. These are one type of file-compression

program, commonly found on CP/M-based bulletin board systems. These use a type of compression called Huffman coding.

There is another less-well-known program called 'lar', which can store a number of files together into a single file, and burst them later. Some people will use 'sq' to squeeze a set of files, and then use 'lar' to put them all together. You might find a file like this, left over from the days before 'arc' was ported to the Amiga. However, 'arc' will compress files better than 'sq'. The 'arc' program has an advantage over 'sq' and 'usq'. Beyond compressing a single file into a file called an 'arc' file, it can store more than one file in an 'arc' file. You can use this to store a number of files together into a single file, to better organize your disks, or to minimize the hassle of downloading a dozen related files, such as programs and their icon files.

Best of all, 'arc' can keep track of the true length of a file stored in the 'arc' file, so the 'chop' program isn't needed. When the 'arc' file is burst of its constituent files, they are all the proper size. 'arc' will always choose the best method of file compression it has. When you add files to an 'arc' file, the program will first analyze the content of the file, in order to choose the best compression scheme, out of the eight methods it knows. The only disadvantage to 'arc' is that it takes time and computer power to 'arc' and un-'arc' files. Of course, it is better to take a few minutes off-line, than to spend extra money downloading a non-'arc'ed file from an expensive network.

If you have another CLI running while you are online, be warned that running 'arc' in the background will rob computer power to the extent that you may lose characters coming in over the modem. 'arc' is a compute-bound process. This kind of program can bring a multitasking computer to its knees, since it will always use most of its time-slice when it has control of the computer.

The 'arc' program is self-documenting. That is, once you have a copy of 'arc', just type 'arc' at a CLI prompt, and it will show you its options, and the form of its command lines. Most commonly, you will use 'arc' to reconstitute a file, after you have downloaded it. If you are downloading an 'arc' file, be sure to name it as such, by including a '.arc' suffix on the name you supply to your terminal program.

For example, if you have downloaded an 'arc' file you called 'mylib.arc', then you would type ARC E MYLIB.ARC to burst the files from the 'arc' file. If you downloaded the file to the RAM: disk, as recommended above, and want to put the files on a disk in drive 'DF1:', type CD DF1:ARC E RAM:MYLIB.ARC and the files in the 'arc' file will be saved to the external disk drive, even though the 'arc' file itself is currently stored in the RAM: drive.

All 'arc' commands are similar to this - first the 'arc' program name, then a single letter, then the name of the 'arc' file to manipulate, and an optional list of files to manipulate. To see a list of the files in an 'arc' file, type ARC L RAM:MYLIB.ARC or use ARC V RAM:MYLIB.ARC for a verbose listing of the files inside, including a chart of the amount of space saved in the compression.



To make a new 'arc' file, composed of one or more files on your disks, use the 'A' option. To add files 'MARTHA' and 'MARY' to an 'arc' file, type ARC A NAMES.ARC MARTHA MARY at a CLI prompt. The 'NAMES.ARC' file will be created in the current directory. If you use the ARC program a lot, then you should copy it to your Workbench disk, in the C directory.

#### RAM: disk bug

Please note that there is a known bug in the RAM: disk driver program, in the Seek() function. Do not do extensive manipulations of 'arc' files in the RAM: disk. I have had no troubles creating new 'arc' files in the RAM: disk, and only adding files to it. If you do anything more extensive, such as the 'D' delete file option in 'arc', move the 'arc' file to disk instead. Presumably, this bug will be fixed in AmigaDOS 1.2.

The 'arc' program is maintained as shareware, as the product of a company in the IBM PC world. It was ported to the Amiga. When you invoke the 'arc' program, the author's name will appear. 'arc' files made on the Amiga can be read by the IBM PC version of the 'arc' program, and vice versa. Of course, executable programs from the PC will not work on the Amiga, and IBM PC text files are formatted somewhat differently than Amiga text files.

PC text file lines are terminated with a carriage-return line-feed sequence, and the Amiga uses only a line-feed character at the end of each line. There are several Amiga programs in the public domain to filter IBM PC text files for use on the Amiga.

#### CIS-B and Kermit

The Compuserve B protocol doesn't add the padding characters to files. It uses 512 byte blocks, which makes it more efficient for transmitting files over national networks. There, the chance of noisy connections are small, and the time for sending characters between computers is large, so the two computers spend more time sending characters, instead of waiting for confirmation of receipt of a packet. However, using CIS-B doesn't free you from the 'chop' dilemma.

Since the person who posted the program could have used Xmodem to upload the file, the file stored on the Compuserve computer has the padded characters, and the Compuserve computer will send them to you. However, if the description of the file says "uploaded with Compuserve B protocol, "you don't have to worry about 'chop'."

The Kermit protocol was named after the famous frog, if you were wondering. 'kermit' also means 'freedom' in Celtic. (In Swahili, 'freedom' is 'uhuru', so in some strange way, Kermit the Frog and Star Trek's Lt. Uhuru are related.) Kermit will send a file correctly, with no padding characters. Few national networks have the Kermit protocol. Compuserve's Executive service has it, but the public network does not. The Source has the Kermit protocol, and there is an Amiga area there, or will be one soon. (Did I just let out a secret?) All Fido Net local bulletin boards have it, and the newest RBBS-PC bulletin board software can optionally support Kermit.

ATTN:  
PASCAL  
USERS

## MODULA-2

the successor to Pascal

- FULL interface to ROM Kernel, Intuition, Workbench and AmigaDOS.
- 32-bit native code implementation with all standard modules.
- Supports transcendental functions and real numbers.
- CODE statement for in-line assembly code.
- Error lister will locate and identify all errors in source code.
- Modula-2 is NOT copy protected.
- 320-page manual

| Benchmarks            | Compile | Link | Execute |
|-----------------------|---------|------|---------|
| Sieve of Eratosthenes | 16      | 32   | 5.3     |
| Null Program          | 14      | 10   | —       |

#### Added features of Modula-2 not found in Pascal

- CASE has an ELSE and may contain subranges
- Dynamic strings of any size
- Machine level interface
  - Bit-wise operators
  - Direct port and Memory access
  - Absolute addressing
  - Interrupt structure
- Programs may be broken up into Modules for separate compilation
- Multi-tasking is supported
- Module version control
- Open array parameters (VAR r: ARRAY OF REALS;)
- Type transfer functions
- Definable scope of object

Pascal and Modula-2 source code are nearly identical. Modula-2 should be thought of as an enhancement to Pascal (they were both designed by Professor Niklaus Wirth).

Regular Version: \$89.95 Developer's Version: \$149.95

The developer's version supplies an extra diskette containing all of the definition module sources, a symbol file decoder, link and load file disassemblers, a source file cross referencer, the kermit file transfer utility and the source code to several of the Amiga Modules.



SOFTWARE, INC.

10410 Markison Road ■ Dallas, Texas 75238 ■ (214) 340-4942  
Telex: 888442 Compuserve Number: 75026,1331

Kermit can send more than one file in a single request, so you could ask for a computer to send you two dozen files, or all the files that begin with 'IFF', and it would send them to your computer, unattended. There are other smart protocols out there, but they are not as popular as Xmodem. One is called Ymodem, an extension of Xmodem. Other protocols exist that let you download files while reading messages, with a corresponding increase in transmission time and cost. But until one of these smarter protocols takes hold, all computers that worry about program file size will be stuck with 'chop'-type problems.

#### AMICUS disks

Last month, I promised to tell the contents of disks 11 through 13. I've got a half-dozen disks of new software to arrange into AMICUS disks, but lost my work to a strange bug that obliterated block 880 of the disk. The 'disksalvage' program saved most of the files, but a dozen or so were lost. I did have backups, but it destroyed several hours of work, organizing the new files.

Beyond the examples of using the AmigaDOS libraries, disk 13 now contains examples of making your own libraries. I wrote these for the FutureSound sound digitizer from Applied Visions. They allow you to load and play IFF sampled sounds from Amiga Basic. This library interfaces C routines to an assembly language core.

Fish disk 25 has the improved game of Hack, which now has true graphics instead of character graphics. Disks 26 and 27 should be here soon.



# Attention:

# Amiga

# Developers

If you are an Amiga developer with an Amazing Amiga product, Amazing Computing can help. To assist in the development of other startup companies, PiM Publications and Amazing Computing have offered advertising rates to Amiga developers at ridiculously low prices.

Our aim is to create a forum for developers in Amazing Computing. We believe we can help the Amiga market produce good, productive software if new development companies have an opportunity to reach their customers without using all of their resources. Therefore, we have offered advertising rates far below what other publications can offer. And our magazine is seen and read by the people Amiga developers want to reach, Amiga owners and Amiga dealers.

Our ridiculously low ad rates are only in effect for a limited time. If you are developing a product for this fast growing market, please take advantage of these rates while we can still offer them. For more information contact:

**Advertising Sales  
Amazing Computing  
PiM Publications  
P.O. Box 869  
Fall River, MA 02722  
617-679-3109**

**Amazing Computing, your Amiga resource.**



# Fun With the Amiga Disk Controller

By Dr. Thom Sterling

The Amiga disk controller is interesting because it provides only the minimum hardware control over the disk drive. The disk controller hardware reads and writes the raw data directly to and from the disk, leaving it up to the programmer to interpret its meaning. Commodore, of course, has provided routines to handle the disks but any dedicated hacker will want to know how to do it themselves.

This article will discuss how the hardware reads from and writes to the disk and how a programmer can write "abnormal" information on a disk, including how to write a total of 1120K bytes of data on a standard Amiga diskette.

In order to write on a diskette the disk controller is given an address in chip memory and a word count. The words pointed to by this address are then sent to the disk drive in a serial fashion with the most significant bit first. The disk drive responds to the data by doing nothing for a "0" bit and by reversing the magnetic field on the write heads for a "1" bit. The controller can be set to send the data at a rate of 1 bit every 2 microseconds or every 4 microseconds. I will only consider the 2 microsecond case.

The diskettes spin at the rate of 300 revolutions per minute, or 5 revolutions per second, which means 200 milliseconds per revolution. Therefore with 2 microseconds per bit, we will have 100,000 bits per revolution and we write: 100,000 bits per track \* 160 tracks per disk = 16,000,000 bits per disk = 2,000,000 bytes per disk. Unfortunately we cannot write just any pattern we choose on the diskette. There are two restrictions: One, there cannot be two flux reversals in a row (so we cannot write two "1"s in a row); two, there cannot be more than 4 "0"s in a row.

The first restriction is because the magnetic field cannot be changed rapidly enough to recognize two reversals close together. The second restriction is because the drive uses the flux transitions to measure the speed at which the drive is rotating and adjusts the clock to compensate for speed variations. The standard method for adapting to these restrictions is called MFM encoding. MFM encoding converts each bit of real data into 2 bits of storage data. A 1 becomes 01 and a 0 becomes 10 or 00 depending on the previous bit: if the previous bit is a 1 then 00 is used, if it is a 0 then 10 is used. So:

AB --> XACB

00 --> ?010  
01 --> ?001  
10 --> 0100  
11 --> 0101.

Logically, between each two bits A and B we insert a bit C, such that:

$$C = \sim(A | B) = \text{NOT} (A \text{ OR } B)$$

Notice the second bit is always the same as the data bit. Restriction one is satisfied because 01 is followed by 01 or 00, and 11 is not allowed, and 10 must be preceded by 00 or 10. Restriction two is satisfied because the most zeroes we can get in a row is 3. (In the pattern 101 --> 010001).

On the Amiga, the conversion of the real data to the stored data is done by software. On many computers, e.g. the IBM PC, this conversion is done by hardware. Luckily, the Amiga has the blitter. Using the blitter, the data can be converted to the MFM form very quickly. Since the stored data has two bits for every real bit, the MFM can be formed more easily if we split the real data into two sets, the even bits and the odd bits. The odd bits are every other bit starting with the first. The evens are all the others. The even and odd bits are stored separately on the disk. This simplifies the conversion because the bit leaves holes in the data in which we can insert the extra bits required by MFM. For example, the data:

01100101 --> evens x1x0x1x1 and odds x0x1x0x0  
--> evens x1000101 and odds x0010010

The encoding for each block can be done in three passes of the blitter. Step one is to find the new bits that will be inserted in between the real data bits. This is done for both even and odd bits at once. The minterm is a blitter technicality:

Blitter input A -- real data shifted right two bits  
B -- real data  
C -- none

Blitter output D --  $\sim(A | B)$  minterm = \$11

Step two merges the even real data with the results from step one:

Blitter input A -- step one D shifted left one bit  
B -- real data  
C -- \$5555 = 0101010101010101

Blitter output D --  $(B \& C) | (A \& \sim C)$  minterm = \$CA

Step three merges the odd real data with the results from step one:

Blitter input A -- real data shifted right one bit  
B -- step one D  
C -- \$5555 = 0101010101010101

Blitter output D --  $(A \& C) | (B \& \sim C)$  minterm = \$AC



Some bits on the ends will still need to be adjusted. The blitter time to do the encoding can be estimated by the number of memory references needed: 11 memory cycles per word, or about 20 milliseconds/track. Since it takes at least 200 milliseconds, plus head seek time, to write to the disk, this is not a big overhead. When the disk is read the data must be decoded. Decoding is easier than encoding and can be done in one blitter operation.

Blitter input A -- stored data shifted left one

B -- stored data

C -- \$5555 = 01010101010101

Blitter output D -- (A & ~C) | (B & C) minterm= \$CA

This will take about 4 memory references per word, which is about 8 milliseconds per track.

The AmigaDOS format encodes the data by sectors, as described in the ROM Kernal Manual Vol 2. I do not know if Commodore actually uses the blitter the way I described, but I have not found a faster way. By using MFM we store 50,000 bits/track which works out to 1,000,000 bytes per disk. Alas, we cannot even get that much on the disk because we must allow for the possibility that the disk spins a little too fast. If the disk spins too fast and we try to write all 50,000 bits, the last bits will be written over the first bits and we will lose that data. A 1 percent increase in speed will overlap 500 bits. Commodore actually stores 47,888 bits on each track, of which 45,056 are user data. While MFM encoding satisfies the restrictions, there are other encoding schemes which store more data per track, and still satisfy the restrictions.

First, we will note that arbitrarily long allowable bit patterns can be formed by arranging the bit strings 01, 001, 0001 and 00001 in arbitrary orders. We can calculate the number of patterns of a given length that can be made this way recursively. For 2 and 3 there is one pattern, for 4 there are two, 0001 and 0101; for 5 there are three, 00001, 01001, and 00101. In general we can calculate the number of patterns,  $X_n$ , for  $n$  bits by the formula:

$$X_n = X_{n-2} + X_{n-3} + X_{n-4} + X_{n-5}$$

The  $X_{n-2}$  term is from 01 followed by all patterns of length  $n-2$ , the  $X_{n-3}$  term is from 001 followed by all patterns  $n-3$  in length, etc.

$nX_n$

| $nX_n$ |     |
|--------|-----|
| -----  |     |
| 21     |     |
| 3      | 1   |
| 5      | 3   |
| 64     |     |
| 77     |     |
| 8      | 10  |
| 9      | 16  |
| 10     | 24  |
| 11     | 37  |
| 12     | 57  |
| 13     | 87  |
| 14     | 134 |
| 15     | 205 |

These patterns are too restrictive because they all start with 0 and end with 1. We can get acceptable patterns by allowing trailing zeroes. For reasons that will become clear we will allow up to 3 trailing zeroes only. We now define  $P_n$  to be:

$$P_n = X_n + X_{n-1} + X_{n-2} + X_{n-3} = X_{n+2}$$

which is the number of patterns with no trailing zeroes plus the number with a single trailing zero plus the number with two and three trailing zeroes. We now have bit strings with from 1 to 4 leading zeroes and from 0 to 3 trailing zeroes. These strings can be put together into acceptable strings if we adopt the rule that the first bit in each pattern becomes a 1 if both of its neighbors are zeroes. This is the same as MFM for  $n=2$ ,  $P_2 = X_4 = 2$ . The two patterns are 01 and 00. There are two other  $n$ 's of particular interest. For  $n=7$ ,  $P_7 = X_9 = 16$ , and so we can encode every 4 bits of real data into 7 bits of stored data (as compared to 8 bits of stored data with MFM). The patterns are:

?001000?000100?010100?010010  
 ?001010?010001?010101?001001  
 ?000101010100001001000100010  
 0101010010000101010010100101

A "?" will be 1 if the preceding 7 bits end with a 0. These can be assigned to the sixteen 4 bit patterns arbitrarily. The second interesting case is  $n=12$ ,  $P_{12} = X_{14} = 134$ , and we can encode 7 bits of real data into 12 bits of stored data (as opposed to 14 bits with MFM). Performing the encoding for this method is more difficult than the MFM encoding. The 7/12 would be encoded by using the blitter to separate the real data into 7 bit pieces, then using a table to find the corresponding 12 bit code, and finally using the blitter again to merge the 12 bit codes into a continuous block. This process will take about 70-80 milliseconds per track. Decoding will be just as hard.

The blitter must be used to split the stored data into 12 bit pieces, then a table is used to find the corresponding 7 bits, and the 7 bit pieces must be merged together again to recover the real data. Decoding will also take 70-80 milliseconds per track. Using the 7/12 code we could nominally store  $7/12 * 100,000 = 58,333$  bits per track. Subtracting some to prevent overlap we might store 56,192 bits per track which is 7K bytes per track or 1120 Kbytes per disk. If the disk is sectorized in a way similar to Commodore, we could fit 13 sectors per track and have 1040 Kbytes per disk of user data. This is an increase of 18 percent over the 880 Kbytes per disk with MFM encoding.

Thom Sterling works for the University of Michigan Physics department, and Errex, Inc.

•AC•



# The AMICUS Public Domain Software Library

This software is collected from user groups and electronic bulletin boards around the nation. Each disk is nearly full, and is fully accessible from the Workbench. If source code is provided for any program, then the executable version is also present. This means that you don't need the C compiler to run these programs. An exception is granted for those programs only of use to people who own a C compiler.

Note: Each description line below may include something like 'S-O-E-D', which stands for 'source, object file, executable and documentation'. Any combination of these letters indicates what forms of the program are present. Basic programs are presented entirely in source code format.

## AMICUS Disk 1

### ABasic programs: Graphics

3DSolids 3d solids modeling program w/sample data files  
Blocks draws blocks  
Cubes draws cubes  
Durer draws pictures in the style of Durer  
FScape draws fractal landscapes  
Hidden 3D drawing program, w/ hidden line removal  
JPad simple paint program  
Optical draw several optical illusions  
PaintBox simple paint program  
Shuttle draws the Shuttle in 3d wireframe  
SpaceArt graphics demo  
Speaker speech utility  
Sphere draws spheres  
Spiral draws color spirals  
ThreeDee 3d function plots  
Topography artificial topography  
Wheels draws circle graphics  
Xenos draws fractal planet landscapes

### ABasic programs: Tools

AddressBook simple database program for addresses  
CardFile simple card file database program  
Demo multwindow demo  
KeyCodes shows keycodes for a key you press  
Menu run many ABasic programs from a menu  
MoreColors way to get more colors on the screen at once, using aliasing  
shapes simple color shape designer Speakt speech and narrator demo

### ABasic programs: Games

BrickOut classic computer brick wall game  
Othello also known as 'go'  
Saucer simple shoot-em-up game  
Spelling simple talking spelling game  
ToyBox selectable graphics demo

### ABasic programs: Sounds

Entertainer plays that tune  
HAL9000 pretends it's a real computer  
Police simple police siren sound  
SugarPlum plays "The Dance of the Sugarplum Fairies"

### C programs:

ATerm simple terminal program, S-E  
cc aid to compiling with Lattice C  
decvt opposite of CONVERT for cross developers  
Dotty source code to the 'dotty' window demo  
echox unix-style filename expansion, partial S, O-D  
fasterfp explains use of fast-floating point math  
FixDate fixes future dates on all files on a disk, S-E  
freedraw simple Workbench drawing program, S-E  
GfxMem graphic memory usage indicator, S-E  
Grep searches for a given string in a file, with documentation  
ham shows off the hold-and-modify method of color generation

## IBM2Amiga

fast parallel cable transfers between an IBM and an Amiga  
Mandel Mandelbrot set program, S-E  
moire patterned graphic demo, S-E  
objfix makes Lattice C object file symbols visible to Wack, S-E  
quick quick sort strings routine  
raw example sample window I/O  
setlace turns on interlace mode, S-E  
sparks qix-type graphic demo, S-E

### Other executable programs:

SpeechToy speech demonstration  
WhichFont displays all available fonts

### Texts:

68020 describes 68020 speedup board from CSA  
Aliases explains uses of the ASSIGN command  
Bugs known bug list in Lattice C 3.02  
CLICard reference card for AmigaDOS CLI  
CLICommands guide to using the CLI  
Commands shorter guide to AmigaDOS CLI commands  
EdCommands guide to the ED editor  
Filenames AmigaDOS filename wildcard conventions  
HalfBright explains rare graphics chips that can do more colors  
ModemPins description of the serial port pinout  
RAMdisks tips on setting up your RAM: disk  
ROMWack tips on using ROMWack  
Sounds explanation of the Instrument demo sound file format  
Speed refutation of the Amiga's CPU and custom chip speed  
WackCmds tips on using Wack

## AMICUS Disk 2

### C programs:

alb AmigaDOS object library manager, S-E  
ar text file archive program, S-E  
fixobj auto-chops executable files  
shell simple CLI shell, S-E  
sq, usq file compression programs, S-E  
YachtC a familiar game, S-E  
Make a simple 'make' programming utility, S-E  
Emacs an early version of the Amiga text editor, S-E-D

### Assembler programs:

bsearch.asm binary search code  
qsort.asm Unix compatible qsort() function, source and C test program  
setjmp.asm setjmp() code for Lattice 3.02  
SVprintf Unix system V compatible printf() function, O-D  
trees.o Unix compatible tree() function, O-D

(This disk formerly had IFF specification files and examples. Since this spec is constantly updated, the IFF spec files have been moved to their own disk in the AMICUS collection. They are not here.)

## John Draper Amiga Tutorial:

Animate describes animation algorithms  
Gadgets tutorial on gadgets  
Menus learn about Intuition menus

## AMICUS Disk 3

### C programs:

Xref a C cross-reference gen., S-E  
6bitcolor extra-half-bright chip gfx demo, S-E  
Chop truncate (chop) files down to size, S-E  
Cleanup removes strange characters from text files  
CR2LF converts carriage returns to line feeds in Amiga files, S-E  
Error adds compile errors to a C file, S  
Hello window ex. from the RKM, S  
Kermit generic Kermit implementation, flakey, no terminal mode, S-E  
Scales sound demo plays scales, S-E  
SkewB Rubik cube demo in hi-res colors, S-E

### AmigaBasicProgs(dir)

Automata cellular automata simulation  
CrazyEights card game  
Graph function graphing programs  
WitchingHour a game

### ABasic programs:

Casino games of poker, blackjack, dice, and craps  
Gomoku also known as 'othello'  
Sabotage sort of an adventure game

### Executable programs:

Disassem a 68000 disassembler, E-D  
DpSlide shows a given set of IFF pictures, E-D  
Arrange a text formatting program, E-D

### Assembler programs:

Argoterm a terminal program with speech and Xmodem, S-E

## AMICUS Disk 4 Files from the original Amiga Technical BBS

Note that some of these files are old, and refer to older versions of the operating system. These files came from the Sun system that served as Amiga technical support HQ for most of 1985. These files do not carry a warranty, and are for educational purposes only. Of course, that's not to say they don't work.

Complete and nearly up-to-date C source to 'image.ed', an early version of the Icon Editor. This is a little flaky, but compiles and runs.

An Intuition demo, in full C source, including files: demomenu.c, demomenu2.c, demoreq.c, getascil.c, idemo.c, idemo.guide, idemo.make, idemo.all.h, nodos.c, and bwrite.c

addmem.c add external memory to the system



|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>bobtest.c<br/>consoleIO.c<br/>creaport.c<br/>creatdi.c<br/>creatask.c<br/>diskio.c<br/>dotty.c<br/>dualplay.c<br/>flood.c<br/>freemap.c<br/>gettools.c<br/>gfxmem.c<br/>hello.c<br/>inputdev.c</p> <p>joystik.c<br/>keybd.c<br/>layertes.c<br/>mousport.c<br/>ownlib.c<br/>ownlib.asm</p> <p>paratest.c<br/>seritest.c<br/>serisamp.c<br/>prinintr.c<br/>prtbse.h<br/>regintec.c<br/>setlace.c<br/>setparallel.c<br/>SetSerial.c</p> <p>singplay.c<br/>speechtoy.c<br/>timedely.c<br/>timer.c<br/>timstuf.c<br/>WhichFont.c</p> <p>process.i and prtbse.i assembler include files:<br/>autogrstr.txt</p> <p>consoleIO.txt<br/>diskfont.txt<br/>fullfunc.txt<br/>inputdev.txt</p> <p>License information on Workbench distribution license printer pre-release copy of the chapter on printer drivers, from RKM 1.1 v11fd.txt 'diff' of .td file changes from version 1.0 to 1.1 v28v1.diff 'diff' of include file changes from version 28 to 1.0</p> <p><b>AMICUS Disk 5. Files from the Amiga Link / Amiga Information Network</b></p> <p>Note that some of these files are old, and refer to older versions of the operating system. These files are from Amiga Link. For a time, Commodore supported Amiga Link, aka AIN, for online developer technical support. It was only up and running for several weeks. These files do not carry a warranty, and are for educational purposes only. Of course, that's not to say they don't work.</p> <p><b>A demo of Intuition menus called 'menudemo', in C source</b></p> <p>whereis.c find a file searching all subdirectories<br/>bobtest.c BOB programming example<br/>sweep.c sound synthesis example</p> <p><b>Assembler files:</b><br/>mydev.asm sample device driver<br/>mylib.asm sample library example<br/>mylib.i<br/>mydev.i<br/>asm supp.i<br/>macros.i assembler include files:</p> <p><b>Texts:</b><br/>amigatricks tips on CLI commands<br/>extdisk external disk specification<br/>gameport game port spec<br/>parallel parallel port spec<br/>serial serial port spec<br/>v1.1.update list of new features in version 1.1<br/>v1.1h.txt 'diff' of include file changes from version 1.0 to 1.1</p> | <p>example of BOB use<br/>console IO example<br/>create and delete ports<br/>create standard I/O requests<br/>creating task examples<br/>example of track read and write<br/>source to the 'dotty window' demo<br/>dual playfield example<br/>flood fill example<br/>old version of 'freemap'<br/>tools for VSprites and BOBs<br/>graphic memory usage indicator<br/>window example from RKM<br/>adding an input handler to the input stream<br/>reading the joystick<br/>direct keyboard reading<br/>layers examples<br/>test mouse port</p> <p>example of making your own library with Lattice<br/>tests parallel port commands<br/>tests serial port commands<br/>example of serial port use<br/>sample printer interface code<br/>printer device definitions<br/>region test program<br/>source to interlace on/off program<br/>set the attributes of the parallel port<br/>set the attributes (parity, data bits) of the serial port<br/>single playfield example<br/>source to narrator and phonetics demo<br/>simple timer demo<br/>exec support timer functions<br/>more exec support timer functions<br/>loads and displays all available system fonts</p> <p>warnings of deadlocks with autorequesters<br/>copy of the RKM console I/O chapter<br/>warning of disk font loading bug<br/>list of #defines, macros, functions<br/>preliminary copy of the input device chapter</p> | <p>Files for building your own printer drivers, including dospecial.c, epsontdata.c, init.asm, printer.c, printer.link, printertag.asm, render.c, and wait.asm. This disk does contain a number of files describing the IFF specification. These are not the latest and greatest files, but remain here for historical purposes. They include text files and C source examples. The latest IFF spec is elsewhere in this library.</p> <p><b>AMICUS Disk 6. IFF Pictures</b></p> <p>This disk includes the DPSlide program, which can view a given series of IFF pictures, and the 'showpic' program, which can view each file at the click of an icon, and the 'saveilbm' program, to turn any screen into an IFF picture. The pictures include a screen from ArticFox, a Degas dancer, the guys at Electronic Arts, a gorilla, horses, King Tut, a lighthouse, a screen from Marble Madness, the Bugs Bunny Martian, a still from an old movie, the Dire Straits moving company, a screen from Pinball Construction Set, a TV newscaster, the PaintCan, a world map, a Porsche, a shuttle mission patch, a tyrannosaurus rex, a planet view, a VISA card, and a ten-speed.</p> <p><b>AMICUS Disk 7. DigiView HAM demo picture disk</b></p> <p>This disk has pictures from the DigiView hold-and-modify video digitizer. It includes the ladies with pencils and lollypops, the young girl, the bulldozer, the horse and buggy, the Byte cover, the dictionary page, the robot and Robert. This includes a program to view each picture separately, and all together as separate, slidable screens.</p> <p><b>AMICUS Disk 8</b></p> <p><b>C programs:</b><br/>Browse view text files on a disk, using menus<br/>S-E-D<br/>Crunch removes comments and white space from C files, S-E<br/>IconExec EXECUTE a series of commands from Workbench S-E<br/>PDScreen<br/>Dump dumps Rasport of highest screen to printer<br/>SetAlternate sets a second image for an icon, when clicked once S-E<br/>SetWindow makes windows for a CLI program to run under Workbench S-E<br/>SmallClock a small digital clock that sits in a window menu bar<br/>Scrimper the screen printer in the fourth Amazing Computing, S-E</p> <p><b>Amiga Basic Programs:</b><br/>(Note: Many of these programs are present on AMICUS Disk 1. Several of these were converted to Amiga Basic, and are included here.)<br/>AddressBook a simple address book database<br/>Ball draws a ball<br/>Cload program to convert Compuserve hex files to binary, S-D<br/>Clue the game, Intuition driven<br/>ColorArt art drawing program<br/>DeluxeDraw the drawing program in the 3rd issue of Amazing Computing, S-D<br/>Eliza conversational computer psychologist<br/>Othello the game, as known as 'go'<br/>RatMaze 3D ratmaze game<br/>ROR boggling graphics demo<br/>Shuttle draws 3D pictures of the space shuttle<br/>Spelling simple spelling program<br/>YoYo wierd zero-gravity yo-yo demo, tracks yo-to the mouse</p> <p><b>Executable programs:</b><br/>3Dcube Modula-2 demo of a rotating cube<br/>AltIcon sets a second icon image, displayed when the icon is clicked<br/>AmigaSpell a slow but simple spelling checker, E-D<br/>arc the ARC file compression program, must-have for telecom, E-D<br/>Bertrand graphics demo<br/>disksalvage a program to rescue trashed disks, E-D<br/>KwikCopy a quick but nasty disk copy program:</p> | <p>ignores errors, E-D<br/>lists hunks in an object file E-D<br/>saves any screen as an IFF picture E-D ??<br/>shareware screen dump program, E only version 2.0, term program, Xmodem E-D</p> <p><b>Texts:</b><br/>LatticeMain tips on fixing _main.c in Lattice<br/>GDSkDrive make your own 5 1/4 drive<br/>GuruMed explains the Guru numbers<br/>Lat3.03bugs bug list of Lattice C version 3.03<br/>MForgeRev user's view of the MicroForge hard drive<br/>PrintSpooler EXECUTE-based print spooling program</p> <p><b>.BMAP files:</b><br/>These are the necessary links between Amiga Basic and the system libraries. To take advantage of the Amiga's capabilities in Basic, you need these files. BMAPs are included for 'clst', 'console', 'diskfont', 'exec', 'icon', 'intuition', 'layers', 'mathfp', 'mathieedoubas', 'mathieesingbas', 'mathtrans', 'potgo', 'timer' and 'translator'.</p> <p><b>AMICUS Disk 9</b></p> <p><b>Amiga Basic Programs:</b><br/>FlightSim simple flight simulator program<br/>HuePalette explains Hue, Saturation, and Intensity<br/>Requester ex. of doing requesters from Amiga Basic<br/>ScrollDemo demonstrates scrolling capabilities<br/>Synthesizer sound program<br/>WorldMap draws a map of the world</p> <p><b>Executable programs:</b><br/>Boing! latest Boing! demo, with selectable speed, E<br/>Brush2C converts an IFF brush to C data instructions, initialization code, E<br/>Brush2Icon converts IFF brush to an icon, E<br/>Dazzle graphics demo, tracks to mouse, E<br/>DeciGEL assembler program for stopping 68010 errors, S-E-D<br/>Klock menu-bar clock and date display, E<br/>life the game of life, E<br/>TimeSet Intuition-based way to set the time and date, E<br/>MEMacs another Emacs, more oriented to word processing, S-E-D<br/>MyCLI a CLI shell, works without the Workbench, S-E-D</p> <p><b>Texts:</b><br/>FnctnKeys explains how to read function keys from Amiga Basic<br/>HackerSin explains how to win the game 'hacker'<br/>lst68010 guide to installing a 68010 in your Amiga<br/>PrinterTip tips on sending escape sequences to your printer<br/>StartupTip tips on setting up your startup-sequence file<br/>XfmrReview list of programs that work with the Transformer</p> <p><b>Printer Drivers:</b><br/>Printer drivers for the Canon PJ-1080A, the C Itoh Prowriter, an improved Epson driver that eliminates streaking, the Epson LQ-800, the Gemini Star-10, the NEC 8025A, the Okidata ML-92, the Panasonic KX-P10xx family, and the Smith-Corona D300, with a document describing the installation process.</p> <p><b>AMICUS Disk 10. Instrument sound demos</b></p> <p>This is an Icon-driven demo, circulated to many dealers. It includes the sounds of an acoustic guitar, an alarm, a banjo, a bass guitar, a boink, a callopie, a car horn, claves, water drip, electric guitar, a flute, a harp arpeggio, a kickdrum, a marimba, a organ minor chord, people talking, pigs, a pipe organ, a Rhodes piano, a saxophone, a sitar, a snare drum, a steel drum, bells, a vibraphone, a violin, a wailing guitar, a horse whinny, and a whistle.</p> |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|



# Fred Fish disks, 1 through 25

## Fred Fish Disk 1:

amigademo Graphical benchmark for comparing amigas.  
amigaterm simple communications program with Xmodem  
balls simulation of the "kinetic thingy" with balls on strings  
colorful Shows off use of hold-and-modify mode.  
dhystone Dhystone benchmark program.  
dotty Source to the "dotty window" demo on the Workbench disk.  
freedraw A small "paint" type program with lines, boxes, etc.  
gad John Draper's Gadget tutorial program  
gbxmem Graphical memory usage display program  
halfbrite demonstrates "Extra-Half-Brite" mode, if you have it  
hello simple window demo  
latftp accessing the Motorola Fast Floating Point I library from C  
palette Sample program for designing color palettes.  
trackdisk Demonstrates use of the trackdisk driver.  
requesters John Draper's requester tutorial and example program.  
speech Sample speech demo program. Stripped down "speechtoy".  
speechtoy Another speech demo program.

## Fred Fish Disk 2:

alib Object module librarian.  
cc Unix-like frontend for Lattice C compiler.  
dbug Macro based C debugging package.  
make Machine independent.  
make2 Subset of Unix make command.  
microemacs Another make subset command.  
portar Small version of emacs editor, with macros, no extensions  
xrf Portable file archiver.  
xrf DECUS C cross reference utility.

## Fred Fish Disk 3:

gothic Gothic font banner printer.  
roff A "roff" type text formatter.  
tf A very fast text formatter  
clorth A highly portable forth implementation. Lots of goodies.  
xlisp Xlisp 1.4, not working correctly.

## Fred Fish Disk 4:

banner Prints horizontal banner  
bgrap A Boyer-Moore grep-like utility  
bison GNU Unix replacement 'yacc', not working.  
bm Another Boyer-Moore grep-like utility  
grep DECUS grep  
kermit simple portable Kermit with no connect mode.  
MyCLI Replacement CLI for the Amiga. Version 1.0  
mandel A Mandelbrot set program, by Robert French and RJ Mical

## Fred Fish Disk 5:

cons Console device demo program with supporting macro routines.  
freemap Creates a visual diagram of free memory  
input.dev sample input handler, traps key or mouse events  
joystick Shows how to set up the gameport device as a joystick.  
keyboard demonstrates direct communications with the keyboard.  
layers Shows use of the layers library  
mandelbrot IFF Mandelbrot program  
mouse hooks up mouse to right joystick port  
one.window console window demo  
parallel Demonstrates access to the parallel port.  
printer opening and using the printer, does a screen dump, not working  
print.support Printer support routines, not working.  
proctest sample process creation code, not working  
region demos split drawing regions  
samplefont sample font with info on creating your own  
serial Demos the serial port

## singlePlayfield

Creates 320 x 200 playfield  
speechtoy latest version of cute speech demo  
speechdemo simplified version of speechtoy, with IO requests  
text.demo displays available fonts  
timer demos timer.device use  
trackdisk demos trackdisk driver

## Fred Fish Disk 6:

compress like Unix compress, a file squeezer  
dadc analog clock impersonator  
microemacs upgraded version of microemacs from disk 2  
mult removes multiple occurring lines in files  
scales demos using sound and audio functions  
setparallel Allows changing parallel port parameters  
setserial Allows changing serial port parameters.  
sortc quicksort based sort program, in C  
stripc Strips comments and extra whitespace from C source

## Fred Fish Disk 7:

This disk contains the executables of the game Hack, version 1.0.1.

## Fred Fish Disk 8:

This disk contains the C source to Hack on disk 7.

## Fred Fish Disk 9:

moire Draws moire patterns in black and white  
MVP-FORTH Mountain View Press Forth, version 1.00.03A. A shareware version of FORTH from Fantasia Systems.  
proff a more powerful text formatting program  
setlace Program to toggle interlace mode on and off.  
skewb a rubik's cube type demo  
sparks moving snake Graphics demo

## Fred Fish Disk 10:

conquest An interstellar adventure simulation game  
dehex convert a hex file to binary  
filezap Patch program for any type of file.  
fixobj Strip garbage off Xmodem transferred files.  
lff Routines to read and write lff format files.  
ld simple directory program  
ls Minimal UNIX ls, with Unix-style wildcarding, in C  
sq.usq file squeeze and unsqueeze  
trek73 Star Trek game  
yachtc Dice game.

## Fred Fish Disk 11:

dpslide slide show program for displaying IFF images with miscellaneous pictures

## Fred Fish Disk 12:

amiga3d Shows a rotating 3 dimensional solid "Amiga sign".  
ArgoTerm a terminal emulator program, written in assembler  
arrow3d Shows a rotating 3 dimensional wire frame arrow.  
ld4 directory listing program  
IconExec two programs for launching programs from Workbench that presently only work under CLI.  
SetAlternate Makes an icon show a second image when clicked once  
StarTerm terminal emu  
lator, with ASCII Xmodem, dialer, more.

## Fred Fish Disk 13:

A Bundle of Basic programs, including:

Jpad toybox ezspeak mandelbrot  
xmodem 3dsolids addbook algebra  
ror amgseq1 amiga-copy band  
bounce box brickout canvas  
cardfi circle colorcircles Copy  
cubes1 cutpaste date dogstar

dragon draw dynamictriangle  
Eliza ezterm filibuster fractal  
fscap gomoku dart haiku  
hal9000 halley hauntedM hidden  
join loz mandel menu  
minipaint mouse Orthello patch  
pena pinwheel gbox random-circles  
Readme rgb rgbtest Rord  
sabotage salestalk shades shapes  
shuttle sketchpad spaceart speak  
speech speecheasy spell sphere  
spiral striper superpad supshr  
talk terminal termtest tom  
topography triangle wheels xenos  
xmostriper  
(note: some programs are Abasic, most are Amigabasic, and some programs are presented in both languages)

## Fred Fish Disk 14:

amiga3d update of #12, includes C source to a full hidden surface removal and 3D graphics  
beep Source for a function that generates a beep sound  
dex extracts text from within C source files  
dimensions demonstrates N dimensional graphics  
filezap update of disk 10, a file patch utility  
gbxmem update of disk 1, graphic memory usage indicator  
gl converts IFF brush files to Image struct, in C text.  
pdlterm simple ANSI VT100 terminal emulator, in 80 x 25 screen  
shell simple Unix 'csh' style shell  
termcap mostly Unix compatible 'termcap' implementation.

## Fred Fish Disk 15:

Blobs graphics demo, like Unix 'worms'  
Clock simple digital clock program for the title bar  
Dazzle An eight-fold symmetry dazzler program. Really pretty!  
Fish double buffered sequence cycle animation of a fish  
Monopoly A really nice monopoly game written in Abasic.  
OkidataDump Okidata ML92 driver and WorkBench screen dump program.  
Polydraw A drawing program written in Abasic.  
Polyfractals A fractal program written in Abasic.

## Fred Fish Disk 16:

Complete copy of the latest developer IFF disk

## Fred Fish Disk 17:

The NewTek Digi-View video digitizer HAM demo disk

## Fred Fish Disk 18:

AmigaDisplay dumb terminal program with bell, selectable fonts  
Ash Prerelease C Shell-like shell program, history, loops, etc.  
Browser wanders a file tree, displays files, all with the mouse  
MC68010 docs on upgrading your Amiga to use a 68010  
Multidim rotate an N dimensional cube with a joystick  
PigLatin SAY command that talks in Pig Latin  
Scripper Screen image printer  
Xlisp 1.6 source, docs, and executable for a Lisp interpreter.

## Fred Fish Disk 19:

BlackJack text-oriented blackjack game  
JayMinerSlides Slides by Jay Miner, Amiga graphics chip designer, showing flowchart of the Amiga internals, in 640 x 400.  
Keymap\_Test test program to test the keymapping routines  
LockMon Find unclosed file locks, for programs that don't clean up.



# Need AMIGA Software?



## Try The Public Domain.....

Amazing Computing™ has vowed, from our begining, to amass the largest selection of Public domain software in the Amiga Community, and with the help of John Foust and Fred Fish, we see a great selection of software for both beginners and advanced users.

These Public Domain software pieces are presented by a world of authors who discovered something fun or interesting on the Amiga and then placed their discoveries in the Public Domain for all to enjoy. You are encouraged to copy and share these disks and programs with your friends, customers and fellow user group members!

The disk are very affordable!

Amazing Computing™ subscribers.....\$6.00 per disk.  
Non subscribers.....\$7.00 per disk

This is extremely reasonable for disks with almost 800K of information and programs. If you agree, please send check or money order to:

PiM Publications Inc.  
P.O. Box 869  
Fall River, MA 02722

Please allow 4 to 6 weeks for delivery

Amazing Computing™: Your resource to the Commodore Amiga

### Fred Fish Disk 20:

AmigaToAtari converts Amiga object code to Atari format  
DiskSalv program to recover files from a trashed AmigaDOS disk.  
Hash example of the AmigaDOS disk hashing function  
Hd Hex dump utility ala Computer Language magazine, April 86  
MandelBrot Mandelbrot contest winners  
MultiTasking Tutorial and examples for Exec level multitasking  
Pack strips whitespace from C source  
PortHandler sample Port-Handler program that performs. Shows BCPL environment clues.  
Random Random number generator in assembly, for C or assembler.  
SetMouse2 sets mouse port to right or left port.  
SpeechTerm terminal emulator with speech capabilities, Xmodem  
TxEd Demo editor from Microsmiths Charlie Heath

### Fred Fish Disk 21

This is a copy of Thomas Wilcox's Mandelbrot Set Explorer disk. Very good!

### Fred Fish Disk 22

This disk contains two new "strains" of microemacs.

Lemacs version 3.6 by Daniel Lawrence. For Unix V7, BSD 4.2, Amiga, MS-DOS, VMS. Uses Amiga function keys, status line, execute, startup files, more.

Pemacs By Andy Poggio. New features include <ALT> keys as Meta keys, mouse support, higher priority, backup files, word wrap, function keys.

### Fred Fish Disk 23

Disk of source for MicroEmacs, several versions for most popular operating systems on micros and mainframes. For people who want to port MicroEmacs to their favorite machine.

### Fred Fish Disk 24:

Conques interstaller adventure simulation game  
Csh update to shell on Disk 14, with built in commands, named variables, substitution.  
Modula-2 A pre-release version of the single pass Modula-2 compiler originally developed for Macintosh at ETHZ. This code was transmitted to the AMIGA and is executed on the AMIGA using a special loader. Binary only.

### Fred Fish Disk 25

Graphic Hack  
A graphic version of the game on disks 7 and 8

### To Be Continued.....

### In Conclusion

To the best of our knowledge, the materials in this library are freely redistributable. This means that they have met one or more of the following conditions:

- (1) The materials contains explicit copyright notices permitting redistribution.
- (2) The materials were posted to a publically accessible electronic bulletin board and did not contain any copyright notice. (Such materials will be removed if it is subsequently shown that copyright notices were illegally removed.)
- (3) The materials were posted to a widely disseminated electronic network (such as usenet), thus implying that their author/poster intended them to be freely distributed. This applies only if they contain no notice limiting distribution.
- (4) The materials contain an explicit notice placing them in the public domain. This is not the same as condition (1).



BECAUSE YOU'RE GOING TO BUY ONLY ONE  
DISK BACKUP UTILITY FOR YOUR



MAKE IT

# MARAUDER®

## MARAUDER IS EVERYTHING YOU EXPECT

Marauder works on any standard Amiga with 256K of memory and one disk drive.

Marauder backs up most copy-protected software titles.

Marauder's user interface is clear-cut, easy to understand, and easy to use.

Marauder's manual is written in plain English without confusing technical terms.

Marauder insures that your valuable software investment will be protected against theft, loss and accidental erasure.

## MARAUDER IS EVERYTHING YOU NEED

Marauder takes advantage of Amiga's optional external disk drive and extra RAM if you have them.

Marauder, itself, is not copy protected in any way.

Marauder uses Amiga's advanced graphics and processing power to its fullest advantage.

Marauder automatically formats your backup disk while copying.

Many complicated protection schemes are easily handled by Marauder's parameter-entry mode — two numbers is all it takes.

Current parameters for all titles will be available **FREE** to registered users at any time.

### BEST OF ALL

### MARAUDER is AVAILABLE NOW!!!

You can get Marauder at the  
**SPECIAL INTRODUCTORY PRICE OF**



# ONLY \$39.95



call now (215) 546-1533

(Dealer Inquiries Welcome)



262 South 15th Street • Suite 300 • Philadelphia, PA 19102-3801



## Bridge the communications gap with a new standard of comparison

### MacroModem

*Efficient for novice or expert  
One keystroke does the work of dozens*

- User defined command macros — 36 commands of 35 characters each. A Macro may contain any key code.
- Macro Help on a function key.
- Command macros stored on disk.
- Load a new command macro set while online.
- Xmodem transfers — Check sum or CRC.
- Transmit text from a disk file.
- Capture a terminal session in a disk file.
- Display the terminal capture file while online.
- The 20 most commonly used commands are invoked with the Amiga Function Keys. HELP lists them.
- User Defined Phone Directory — 36 numbers per disk file. Change directories while on line.
- Auto Dial — pick a number from the directory window or enter it from the keyboard.
- Includes MacroModem Editor — a multi-window editor for Macro and Phone files.
- Includes FileFilter — a file copy utility that ends file format problems:
  - Finds the end of Amiga binary files
  - Chops data files to specified length\*
  - Translates Amiga, Mac and IBM text files
  - Expands tabs
  - Inserts or removes form feeds
- Unlimited baud rates from 112 to 262,000.
- User selected serial capture buffer size.
- Two terminal display modes — TTY and ANSI.
- Command mode with multiple commands on a line.
- SHELL command for calling AmigaDOS.
- Multi-window operation.
- NewCLI — Create a new AmigaDOS window when you need it, even during File Transfers.
- Runs From CLI or Workbench.
- Requires 256k and 1 disk drive.

**Price \$69.95**

**Available Now**

Dealer inquiries invited

Kent Engineering & Design  
Box 178, Mottville, NY 13119  
(315) 685-8237



**The bridge to your computing future.**

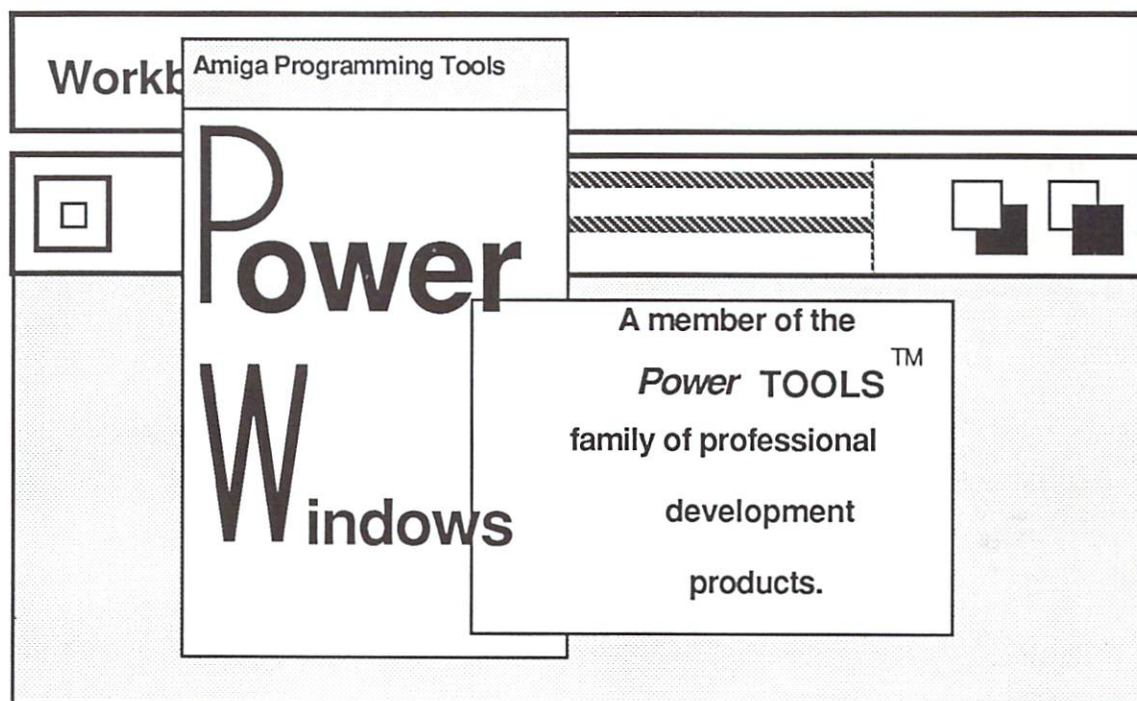
Amiga, IBM, Macintosh are trademarks of Commodore - Amiga, Inc., International Business Machines, and Apple Computer, respectively.

## Index of Advertisers

|                                   |                    |
|-----------------------------------|--------------------|
| Adept Software                    | 59                 |
| Advanced Systems Design Group     | 50                 |
| Amiga Project                     | 64                 |
| Anakin Research                   | 40                 |
| Byte By Byte                      | C IV               |
| Cardinal Software                 | 54,55              |
| Colony Software                   | 12                 |
| Computer West                     | 56                 |
| Commspec Communications           | 24                 |
| Data Reductions Associates        | 51                 |
| Data Research Processing          | 42                 |
| DESKWARE                          | 61                 |
| Digitek                           | 7                  |
| Discovery Software                | 1,79               |
| Eastern Telecom Inc.              | 8                  |
| Felsina Software                  | 31                 |
| Golden Hawk Technology            | 39                 |
| Great Cover-Ups                   | 34                 |
| Image Set                         | 59                 |
| Inovatronics, Inc.                | CIII               |
| Interactive Analytic Node         | 68,69              |
| Jen Day Software                  | 28                 |
| K J Computers                     | 60                 |
| Lattice, Inc                      | 5                  |
| LionHeart                         | 37                 |
| Macro Ware                        | 80                 |
| Meridian Software Inc.            | 38                 |
| Metadigm, Inc.                    | 2                  |
| Michigan Software Distributors    | 53                 |
| Micro Search Inc., Amiga House    | 72                 |
| MicroSmiths, Inc.                 | 16                 |
| Micro-Systems Software Inc.       | 43                 |
| MIDI-DESIGNS                      | 33                 |
| Netch Computer Products           | 19                 |
| Nikol & Company                   | 61                 |
| Pacifico Inc.                     | 62                 |
| PIM Publications                  | 2,10,66,72,78,C II |
| Professional Network Services Co. | 70                 |
| Quality Cottage                   | 18                 |
| SKE Software                      | 36                 |
| Slipped Disk                      | 42                 |
| Software Supermarket              | 65                 |
| TDI Software                      | 71                 |
| Tigress                           | 14                 |
| T&L Products                      | 26                 |
| TPUG                              | 58                 |
| Transtime Technology Co.          | 48                 |
| Westcom Industries                | 27                 |
| ZOXSO                             | 46                 |

•AC•





The **first** interactive Amiga program design tool, *PowerWindows*™ lets you to design fantastic looking windows, menus and gadgets in minutes instead of hours or days! You show this incredible program what you want and it does the rest, generating C or 68000 assembler source code for you to include in your own programs. *PowerWindows* is a structure generator for a machine that **thrives** on structures. With this software package you can:

Pick the exact size and position for your windows **visually**. No more "wait to see what it looks like"; *PowerWindows* knows where your window is and everything else about it!

Design professional looking menus. Add menus, move menus, or delete menus, whatever you want to do with text menus, our program keeps track of them and writes source code letting you duplicate them exactly with simple operating system calls.

Create your own string, integer and boolean gadgets and position them anywhere in your window. *PowerWindows* keeps them from colliding and remembers the type, location and text contents of each one for writing those complex gadget structures.

Best of all, you can keep your designs in a format that can be re-edited, letting you create your favorite type of windows and customize them for each program you write.

#### Order Form

Price for *PowerWindows* is \$89.95, plus \$3.50 for shipping and handling. Texas residents please add 6.125% sales tax to total price.

Name \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_

State \_\_\_\_\_

Zip \_\_\_\_\_

Products ordered \_\_\_\_\_

Payment method: \_\_\_\_\_ MC/Visa \_\_\_\_\_ Check \_\_\_\_\_ Money Order

AC

Card Number \_\_\_\_\_

Name on card \_\_\_\_\_

Signature \_\_\_\_\_

Expiration Date \_\_\_\_\_

Enter total enclosed: \_\_\_\_\_

## INOVATRONICS, INC.

11311 Stemmons Frwy., Suite 7

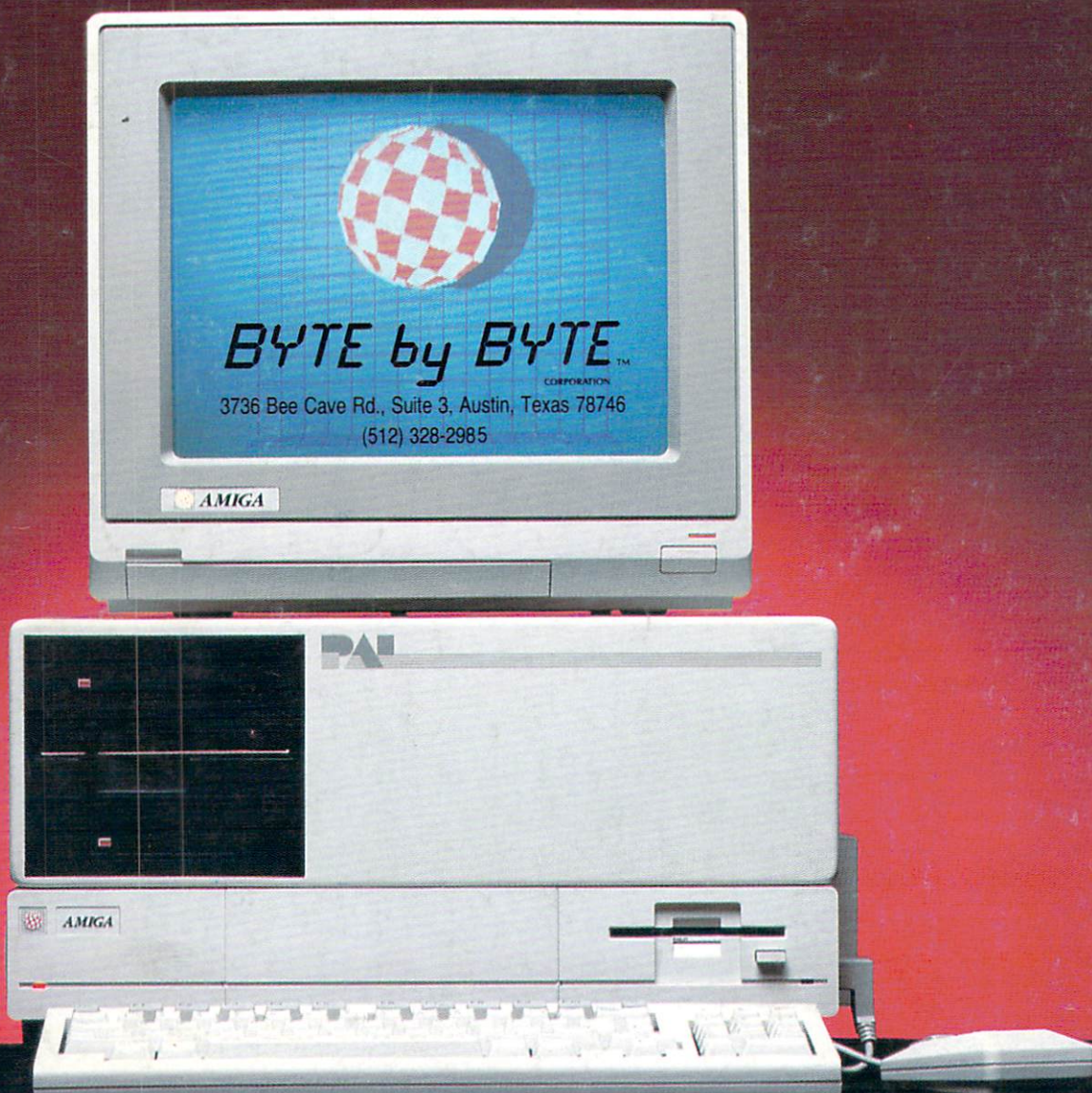
Dallas, TX 75229

214/241-9515



# UNLEASH THE AWESOME POWER OF THE AMIGA!

The PAL is a turnkey expansion chassis that provides the most powerful and cost effective hardware growth path for your AMIGA. Features: High speed direct Amiga DMA controller and hard disk • Five DMA expansion slots • 1 Meg Ram with Clock/Calendar • Room for multiple storage/retrieval devices • 100% compatible with current and future Amigas • 1 to 8 megabyte ram card options • Optional pass through bus connector for further expansion • Optional prototyping card • Future products currently under development



INFOMINDER is an intelligent information resource that provides the user with instantaneous access to reference information stored within the Amiga personal computer.

Fully supports multi-tasking • Fast access by menu or outline • Text capabilities include: Justification, Word Wrap, Multiple character fonts/styles • Information

content completely user definable • Supports combination of TEXT and IFF GRAPHICS • Programmatic interface for context sensitive help • Narration and printing of information • Expand and shrink topics.

INFOMINDER will revolutionize the way we access textual and graphical information. Stop searching and START using the information around you.

Special introductory price \$89.95

WRITE HAND is a general word processor and form letter generator that gives you the most features for your dollars. Developed to meet the special needs of small business, WRITE HAND is easy to learn and easy to use.

WRITE HAND challenges you to compare the following features dollar-for-dollar, feature-for-feature to those of

other word processors on the market today.  
• Extensive on-line HELP service • Form letter generator • Powerful editing capabilities • Formats documents while you edit • Reviews and merges files while you edit • Moves blocks of text and figures of any size • Provides word wrap, bolding and underlining

Make WRITE HAND the tool that moves your business into the productive world of electronic word processing. Suggested retail price \$50.

FINANCIAL PLUS is the affordable way to put your business at your fingertips. FINANCIAL PLUS is the complete accounting solution with five systems in one:  
• General Ledger • Accounts Payable • Accounts Receivable • Payroll • Word Processor

FINANCIAL PLUS is adaptable. You customize each company according to

its size and bookkeeping needs.

An easy-to-read, easy-to-learn users guide provides comprehensive instructions for setting up your own books. Plain-English menus are the system "roadmaps" for both the novice and for the more experienced. Because FINANCIAL PLUS is a totally integrated accounting system, no longer must you purchase individual packages, store entries on separate diskettes, or run confusing transfer programs to obtain complete integration. Suggested retail price \$295.